

2 Один из тестов симуляции также представляет результаты использования стандартного интеграционного метода трапеций второго порядка.

## 7.3.2 Осциллятор на таймере 555

Рис. 7.9 показывает схему базового автоколебательного импульсного генератора на таймере 555. Время заряда конденсатора C1 дается:  $t_c = 0.693(R5 + R6)C1$  секунд, и разряда:  $t_d = 0.693(R6)C1$  секунд. Следовательно, период и частота осцилляции:

$$T = t_c + t_d = 0.693(R5 + 2R6)C1 \text{ секунд, и } f = \frac{1.44}{(R5 + 2R6)C1} \text{ Hz.}$$

Рабочий цикл для выходного графика таймера задается  $D = \frac{R6}{R5 + 2R6}$ .

Рис. 7.10 иллюстрирует график симуляции для автоколебательного осциллятора. Когда резистор R6 шунтируется диодом, конденсатор C1 заряжается через резистор R5 и разряжается через резистор R6. При задании  $R5 = R6$  получается 50и-процентный рабочий цикл<sup>3</sup>, см. рис. 7.11 и 7.12.

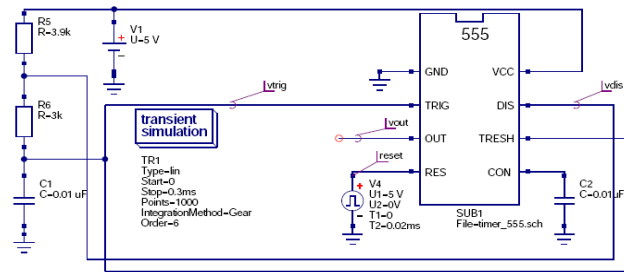


Рис. 7.9. Импульсный автогенератор с таймером 555

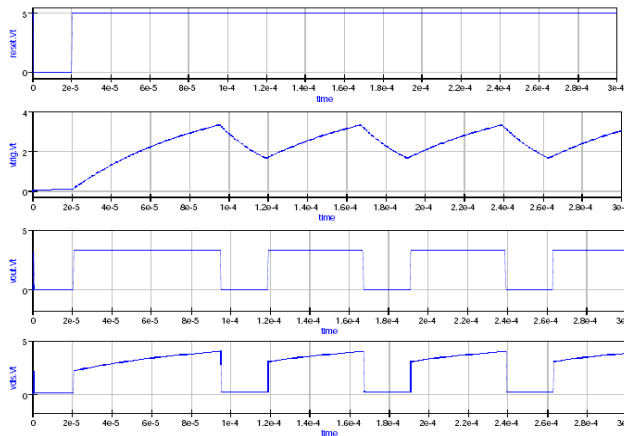


Рис. 7.10. Графики симуляции для базового импульсного автогенератора

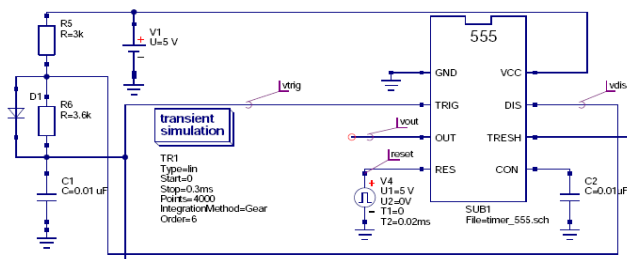


Рис. 7.11. Импульсный автогенератор с 50и-процентом циклом на таймере 555

<sup>3</sup> Значение R6 нужно подобрать, чтобы получить рабочий цикл точно 50и-процентным.

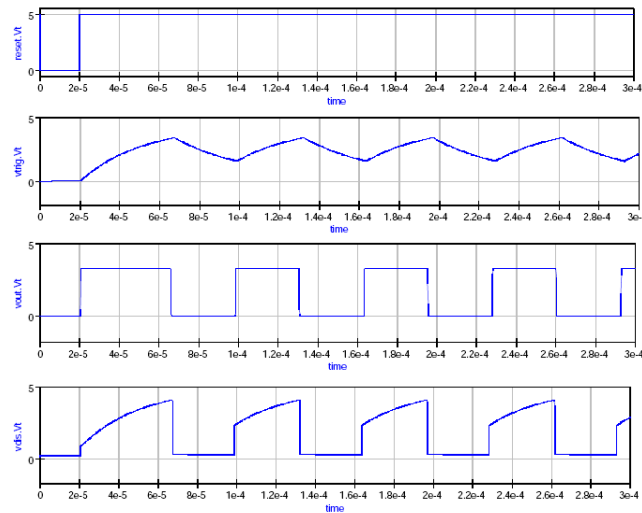


Рис. 7.12. Графический вывод симуляции для 50и-процентного цикла импульсного автогенератора

### 7.3.3 Модификация ширины импульса

Переключение таймера 555 в одно-стабильном режиме с продолжительной последовательностью импульсов позволяет модифицировать ширину выходных импульсов изменением амплитуды сигналов, приложенных к управляющему входу вывода 5 (CON). Пример схемы модуляции ширины импульсов дан на рис. 7.13. На этой схеме компоненты C2, R6 и D1 конвертируют управляющий сигнал таймера 555 в переключающий задним фронтом сигнал. Это можно увидеть на рис. 7.14, который показывает графики переключения, разряда и результирующий выходной сигнал. Управляющий вывод таймера 555 запитан от источника импульсного напряжения. Спецификация управляющего графического вывода была выбрана для генерации сигнала треугольной формы, так что модуляция ширины импульсов может быть ясно показана как изменение амплитуды управляющих сигналов.

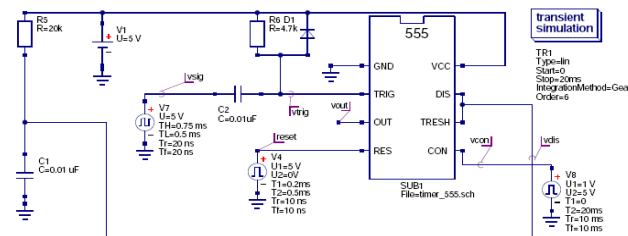


Рис. 7.13. Схема модулятора ширины импульсов на таймере 555

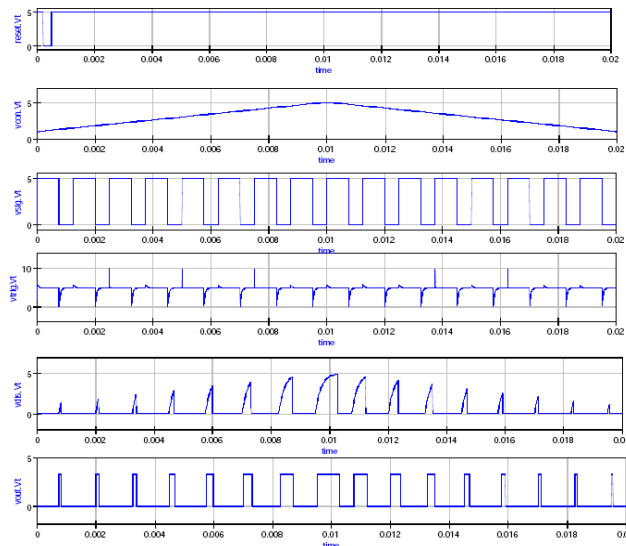


Рис. 7.14. Графики симуляции модулятора ширины импульсов

### 7.3.4 Модуляция позиционного импульса

Модулятор позиционного импульса может быть собран из автоколебательного генератора показанного на рис. 7.9. Модулирующий сигнал приложен ко входу управления, вывод 5 (CON); см. рис. 7.15. Этот сигнал приводит к тому, что позиция импульса варьируется с амплитудой модулирующего сигнала. Типовой набор графиков симуляции для этой схемы показан на рис. 7.16. Это очень трудная для симуляции схема. И это один из случаев, когда трапецидальный метод интеграции работает успешно, тогда как интеграционный метод Gear шестого порядка аварийно завершается<sup>4</sup>. Заметьте, что результаты метода трапеций были получены с использованием 30000 точек, начальный шаг = 0.001 nS, MinStep = 1e-16, MaxIter = 5000, abstol = 10uA и vntol = 10uV.

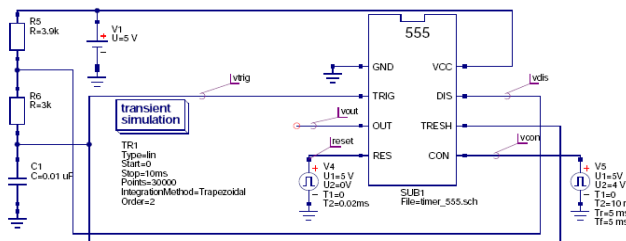


Рис 7.15. Схема модулятора позиционного импульса с таймером 555

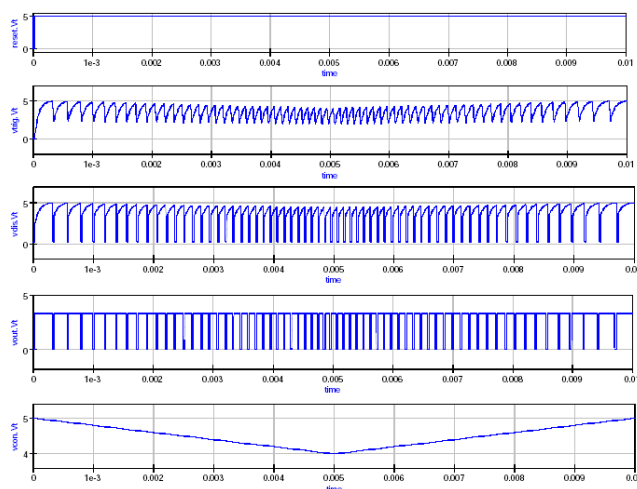


Рис. 7.16. Графические выходы симуляции модулятора позиции импульса, полученные с помощью интеграционного метода трапеций

<sup>4</sup> Симуляция переходного процесса никогда не завершается и может быть прекращена щелчком по клавише прекращения симуляции.

## 7.4 Примеры симуляции с несколькими таймерами 555

Убедившись в последнем разделе, что новая Qucs модель таймера 555 может симулировать стандартные схемы приложений, перечисленные в справочных данных, в этой части руководства представим два следующих, более сложных примера, демонстрирующих, как таймер 555 используется на практике.

### 7.4.1 Генерация последовательной серии импульсов

Частое практическое применение таймер 555 находит в генераторах тактовых импульсов для целей управления. Схема на рис. 7.17 показывает ряд одно-стабильных генераторов импульсов, соединенных последовательно и параллельно. После сброса схемы спадающим фронтом входного импульса  $v_{in}$  включается генерация последовательности импульсов. Длительность каждого одно-стабильного импульса устанавливается внешними конденсаторами от C1 до C4<sup>5</sup>. Спецификация подсхемы одно-стабильного импульсного генератора дана на рис. 7.18. Генератор последовательности импульсов – составная схема с:

60 R образами, 40 C образами, 4 vcvs образами, 1 vdc образами, 8 Idc образами, 2 Vpulse образами, 8 OpAmp образами, 4 Diode образами, 4 BJT образами, 8 Inv образами, 8 NOR образами и 4 OR образами.

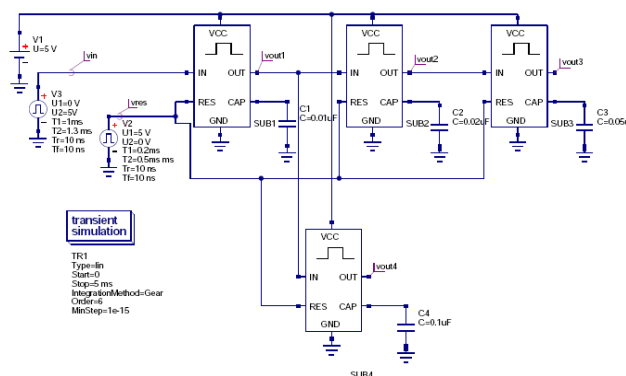


Рис. 7.17. Схема генератора последовательности импульсов

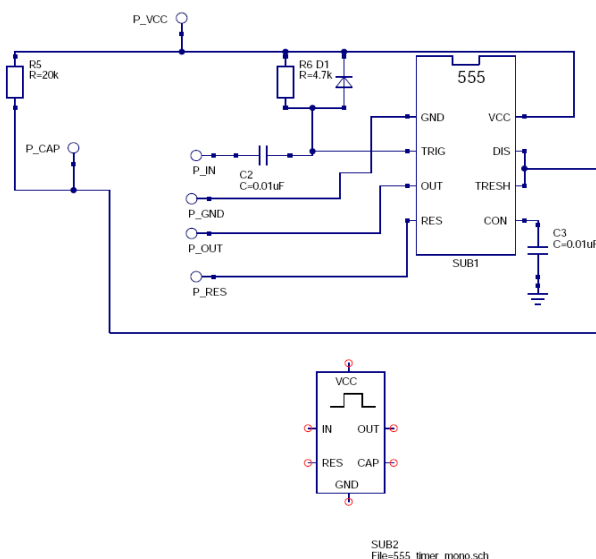


Рис. 7.18. Подсхема одно-стабильного импульсного генератора

<sup>5</sup> Длительность импульсов, устанавливаемая C1 – C4 на рис. 7.17, имеет простой выбор для целей демонстрации и не представляет какой-либо особенной управляющей временной последовательности.

Большое число компонент, и действительная сложность схемы, имеет тенденцию к тому, чтобы время симуляции схемы генератора серии импульсов стало много больше обычных времен записи, когда симулируется единственная цепь таймера 555. Проблемы DC сходимости и ошибки временного шага анализа переходного процесса могут вызвать проблемы из-за разрывности переключения, так что не лишним будет тщательный выбор параметров нелинейных диодов и естественных условий анализа переходных процессов. На рис. 7.18 диод используется для сжимания сигнала переключающего входа таймера 555 до пяти вольт, когда сигнал возрастает выше 5 вольт. Параметры диода Qucs по умолчанию похожи на заданные в SPICE<sup>6</sup>. По умолчанию постоянная эмиссии диода установлена в 1, а последовательный с диодом резистор в ноль Ом. Ни одно из этих значений не представительно для кремниевых диодов. Для кремниевых устройств, в отличие от германиевых диодов,  $n$  нужно задать между, примерно, 1.5 и 2. Аналогично, все диоды имеют некоторые последовательные резисторы, часто в диапазоне от 0.1 до 10 Ом, в зависимости от мощности диода. Для облегчения симуляции эти значения были установлены в  $n = 2$  и  $R_s = 10$  Ом. Рис. 7.19 показывает типичный набор сигналов графического вывода, полученный при симуляции генератора последовательности импульсов. Условия симуляции, выбранные для получения этих результатов: метод интеграции = Gear, порядок = 6,  $\text{initialStep} = 1 \text{ ns}$ ,  $\text{MinStep} = 1\text{e-}15$ ,  $\text{reltol} = 0.001$ ,  $\text{abstol} = 10 \mu \text{ A}$ ,  $\text{vntol} = 10 \mu \text{ V}$ , Solver = CroutLU и  $\text{initialDC} = \text{yes}$ .

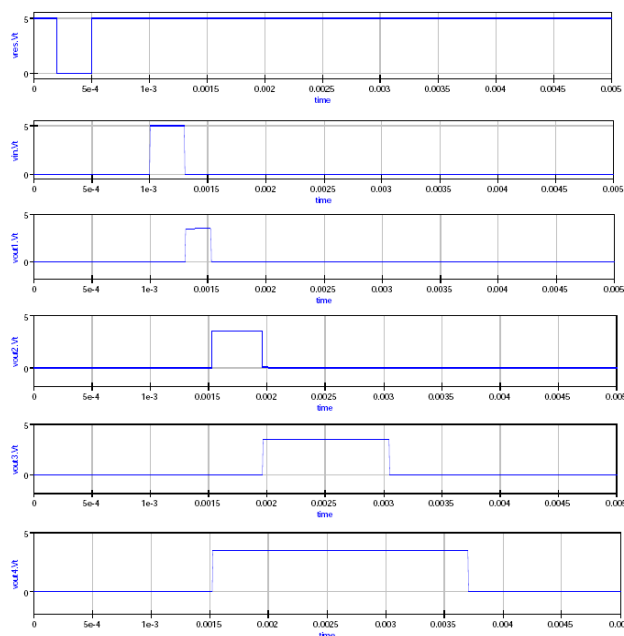


Рис. 7.19. Графики симуляции для моно-стабильного генератора импульсов

## 7.4.2 Схема делителя частоты

Общее требование в обоих режимах, и при разработке цифровых, и смешанных цепей – это деление частоты, где последовательность импульсов высокой частоты, получаемая часто от кварцевого генератора, делится на более низкие частоты<sup>7</sup>. Классический пример деления таких сигналов – использование цепочки триггеров, каждый из которых соединяется как делитель на два. Таймер 555 может также использоваться для деления частоты импульсов<sup>8</sup>. Схема, показанная на рис. 7.20, – базовая схема одно-стабильного режима 555 с последовательностью импульсов приложенной ко входу переключения таймера 555, вывод 2 (TRIG). В предыдущих разделах этих заметок пояснялось, что вход триггерного компаратора 555 чувствителен к уровню сигнала, и переключение происходит, если длительность низкого уровня импульса больше длительности одно-

<sup>6</sup> Предопределенные значения установлены в ранней версии SPICE, возможно, версии 1, и, похоже, не менялись при дальнейшей проработке симулятора.

<sup>7</sup> Часто результирующая частота в диапазоне от 1 до 5 Hz и используется для мигания LED, или каких-то других оптических индикаторов, включено/выключено.

<sup>8</sup> Таймеры 555 обычно более эффективны, чем триггеры в этом применении, поскольку единственное устройство может обеспечить деление больше, чем на два.

стабильного импульса. На рис. 7.20 длительность одно-стабильного импульса 0.22ms и параметр прямоугольного напряжения генератора TL – 0.5ms, который приводит к появлению переключения. Эффекты переключения могут просматриваться на рис. 7.21. Применение таймеров 555 для деления частоты базируется на одно-стабильной схеме, показанной на рис. 7.20 и, следовательно, разработчики схемы должны удостовериться, что переключение имеет место. Иллюстрация рис. 7.22 – это двух-каскадный делитель частоты, где каждый каскад делит последовательность входных импульсов на пять, давая общее деление всей схемы равное двадцати пяти. Выходные графики для этой схемы показаны на рис. 7.23. При разработке схем делителя частоты на таймере 555 хорошо бы сделать, чтобы период таймера 555 был установлен в  $(N-0.5)$  времени периода входной последовательности импульсов<sup>9</sup>, где N – к-т деления и должен быть в диапазоне  $2 \leq N \leq 10$ .

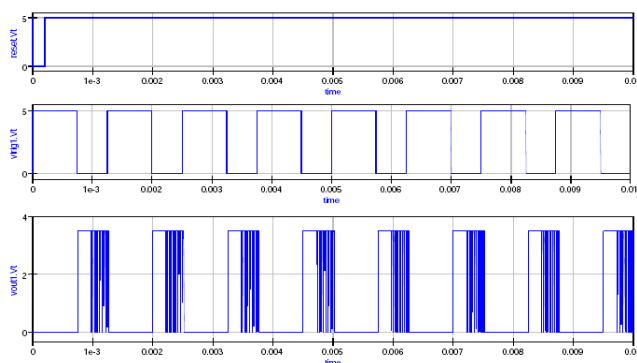


Рис. 7.21. Графика симуляции для схемы рис. 7.20, показано переключение таймера 555

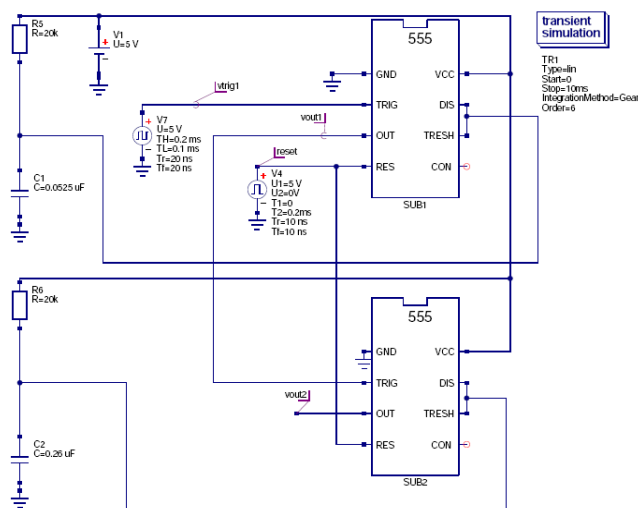
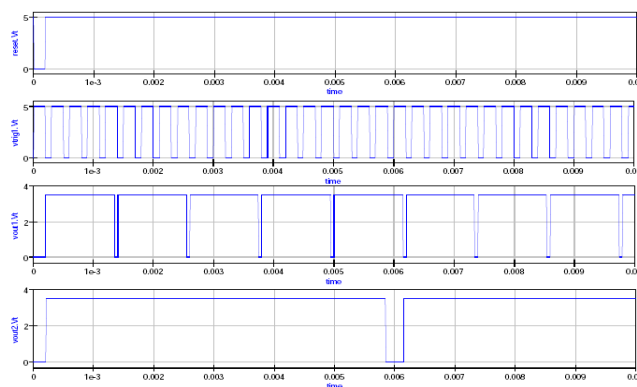


Рис. 7.22. Двух-каскадный делитель частоты на таймере 555



## **7.5 Заключительные замечания**

Разработка модели симуляции для таймера 555 – это интересная задача. В этих заметках руководства сделана попытка описать принципы и технологию макро-моделирования для такого рода задач. Они также демонстрируют, насколько Qucs созрел для того, чтобы стать универсальным симулятором. Новая модель Qucs 555 таймера – это, в значительной мере, первая попытка моей части построения функциональной модели этого сложного устройства. Много больше работы следует сделать в дальнейшем, чтобы улучшить модель таймера 555. Модели таймера 555 с низким энергопотреблением тоже нуждаются во внимании. Много времени потребуется, чтобы универсальная параметризованная подсхема модели для таймера 555 реализовала передачу параметров в подсхемы Qucs и вычисляла значения компонент, используя уравнения. Специальная благодарность Stefan Jahn за его ободрение и множество модификаций, которые он сделал в Qucs, чтобы либо исправить ошибки, либо добавить функциональности на протяжении всей моей работы над этой частью.

## 8 Смещение ВТ транзистора

### 8.1 Графические методы

Вы можете сместить переход биполярного транзистора несколькими способами. Определить же лучший вариант для вашей цели легче всего с помощью графического метода. Смещение активного устройства, такого как биполярный плоскостной транзистор (ВТ), требует, чтобы вы задали DC напряжения и токи устройства. Для оптимизации желаемых результатов нужны разные значения смещения. Например, входное устройство усилителя с низким уровнем шума может иметь наилучшие шумовые показатели при токе коллектора  $50 \mu\text{A}$  и максимальном напряжении коллектор-эмиттер  $5\text{V}$ , тогда как следующий каскад усилителя должен иметь ток коллектора  $20\text{mA}$  и напряжение коллектор-эмиттер  $18\text{V}$  для создания необходимого АС напряжения на выходе. Как только вы определи желаемые условия смещения, вам нужно удостовериться, что они воспроизводимы при некоторых ограничениях, что гарантировало бы согласованное представление.

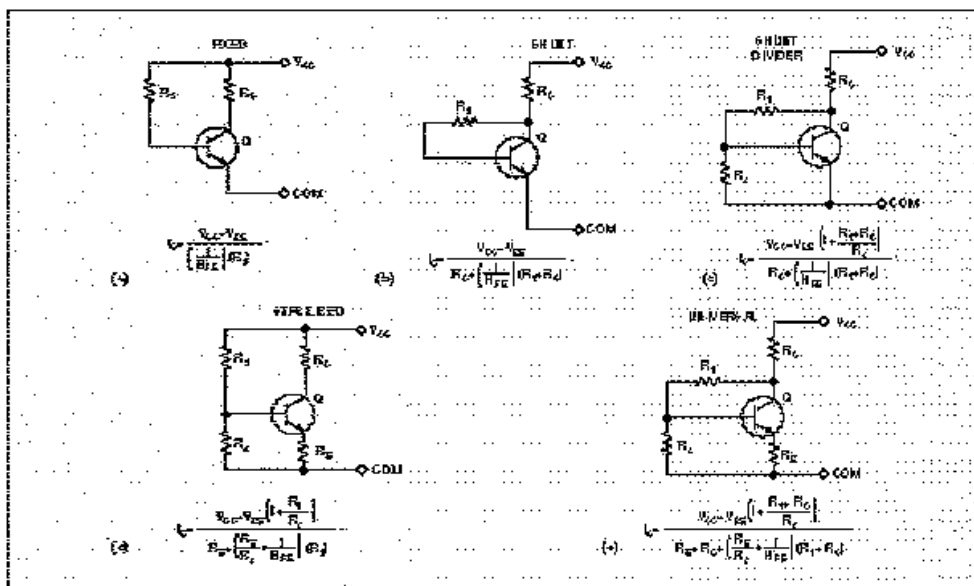


Рис. 8.1. Различные способы обратной связи

Анализ способов смещения для ВТ обычно развивается, с усложнением, из метода фиксированного смещения (см. рис. 8.1), переходя к цепям шунтирования, к цепям стабилизации и т. д. Сюда, обычно, входят шунтирующие делители и универсальные цепи задания смещения. Однако по поводу стабильности смещения цепью шунтирования еще остаются вопросы. Способ подходит в некоторых некритических случаях, но насколько он ухудшает стабилизируемую схему? Разработчиков, как правило, учат, что схема стабилизации – единственная для повторения.

Один из путей анализа стабильности различных методов смещения – использование параметров стабильности, характеризующих изменение тока коллектора из-за изменения HFE (усиление по току) транзистора, ICBO (ток утечки коллектор-база), и VBE. Хотя эти параметры полезны, сравнение цепей смещения и значений резисторов смещения требует утомительных вычислений. Большую пользу принесет визуальный показ сравнительной стабильности разных цепей. Взгляните на уравнения для тока коллектора (IC) рис. 1b, и заметьте, что большинство изменений в IC происходит из-за разницы напряжений, возникающих на R1, по причине разброса HFE. Это приводит к вопросу: если ток через R1 фиксирован, будет ли результатом уменьшение изменений напряжения на R1 и, следовательно, уменьшение изменений IC? Эти соображения приводят к схеме шунтирующего делителя (рис. 1c). Поскольку VBE меняется мало, R2 поддерживает относительно постоянную составляющую тока через R1, при том, что R1 получается с меньшим значением, чем оно было бы без R2. Уравнение для шунтирующего делителя показывает, что меньшее значение R1 в знаменателе приводит к

меньшим изменениям для  $I_C$  из-за изменения  $HFE$ . Однако с  $R_C$  и  $R_2$ ,  $R_1$  в числителе в качестве множителя для  $V_{BE}$ .

Вы можете продолжать рассматривать, насколько сильно каждый из этих факторов влияет на  $I_C$ . Поскольку вы можете воспроизвести все схемы рис. 1 из универсальной цепи (рис. 1e), делая соответствующие резисторы либо бесконечными (открытая цепь), либо нулевыми (замкнутая цепь), та же универсальность относится и к уравнениям. Рассмотрев уравнения цепей в диапазоне параметров и значений смещающих резисторов, вы можете построить график, на котором ось  $Y$  представит изменения в  $I_C$ . Чтобы сделать правильное сравнение цепей, нужен общий параметр, относящийся к смещению, для оси  $X$ . Отношение тока коллектора к току смещения через  $R_1$  работает. Это отношение общее для цепей и отражает то, как изменяется смещение. Для показа реальных условий, данные также включают температурное влияние на  $V_{BE}$  и  $HFE$  в диапазоне от 25 до 75°C и от 3 до 1 приращение для  $HFE$ . Для целей сравнения все цепи используют 10V, как напряжение питания для  $V_{CC}$  при номинальном токе коллектора в 1 mA, с  $HFE$  в 100 и  $V_{BE}$  в 0.60V при 25°C. Вычисление резисторов для 5V  $V_{CE}$  и выбор  $R_E$  для создания 1V на эмиттере дают результаты для графического метода. Модель для температурных эффектов устройства –  $V_{BE}=0.60+0.002(T(actual)-25^{\circ}C)$ , представляющая стандартный 2mV/°C коэффициент для диодов. Вычисления из справочных данных транзистора 2N2222A дают средний температурный к-т для  $HFE$  около 0.58% /°C, который может быть представлен

$$HFE_{Temp} = HFE_{Max} \times [1 + (T(actual)-25^{\circ}C)0.0058] \quad (8.1)$$

Расчет  $I_C$  на минимум  $HFE = 50$  при 25°C и максимум  $HFE = 150$  при 75°C приносит  $HFE_{Temp}$  в 194 и  $V_{BE}$  в 0.50V.

Этот анализ игнорирует эффекты  $I_{CBO}$ . Для номинального тока коллектора в 1 mA и максимальной температуры 75°C вклад  $I_{CBO}$  в  $I_C$  дает несколько процентов, самое большое, для фиксированного смещения и цепей шунтирования смещения на рис. 1a и 1b, и меньше для цепей смещения на рис. 1c, 1d и 1e.

### 8.1.1 Графическое приближение показывает компромисс

Результат анализа появляется, как простое визуальное сравнение стабильности схем при разных типах цепей смещения (рис. 8.2). Используя рисунок, вы можете выбрать тип задания смещения и параметры смещения для получения необходимой стабильности.

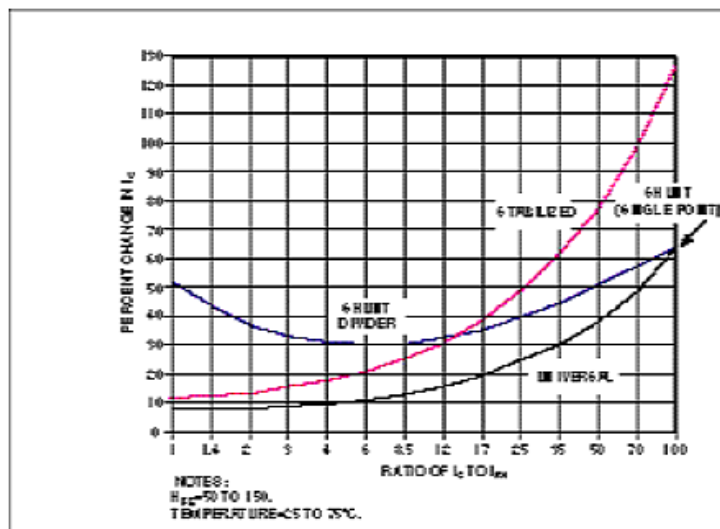


Рис. 8.2. Вы можете сравнить представление цепи смещения BJT с графиком изменения тока коллектора от соотношения тока коллектора и тока через  $R_1$

Горизонтальная ось – это отношение тока коллектора ( $I_C$ ) к току через резистор  $R_1$ . Это отношение смещения приложено ко всем цепям и показывает ток в цепи смещения базы, сравнительно с током коллектора. Так,

отношение равное 1 показывает устойчивую цепь смещения, с таким же током через  $R1$  цепи смещения, что и в цепи коллектора, тогда как отношение 50 показывает, что ток коллектора в 50 раз больше тока через резистор  $R1$  цепи смещения. Поскольку некоторые из этих результатов неожиданны, они дают новый взгляд на отдельные цепи смещения, прежде игнорируемые. Универсальный метод смещения, очевидно, наилучший в группе. Цена, которую вы платите за ее DC стабильность – это уменьшение входного сопротивления из-за отрицательной обратной связи через  $R1$ , вид эффекта Miller на резисторах.  $R1$  уменьшается на  $k$ -т усиления по напряжению плюс 1. Эта обратная связь уменьшает искажения и улучшает полосу пропускания, а также уменьшает выходной импеданс на коллекторе. Если вы не хотите иметь эти AC эффекты обратной связи, вы можете устранить их, разделив  $R1$  на две части и пропустив центр на землю (рис. 8.3). Вы можете улучшить действие этой цепи в отношении смещения, увеличив падение напряжения на  $RE$ , увеличив падение напряжения на резисторе в цепи коллектора или увеличив оба.

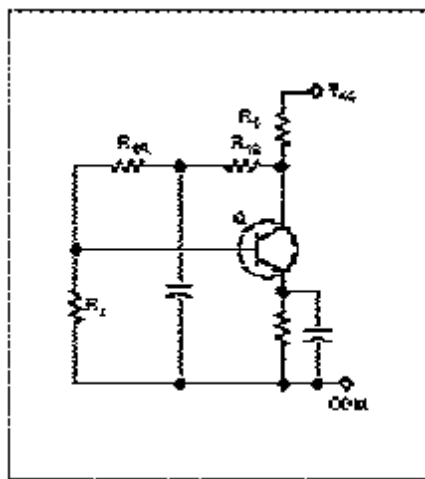


Рис. 8.3. Чтобы устранить AC эффекты обратной связи, разделите  $R1$  и центр пропустите на землю

Цепь стабилизации имеет хорошую стабильность для смещения с отношением не выше 12. Выше этого отношения ее стабильность быстро уменьшается. Стабилизация цепи происходит и от изменений напряжения обратной связи по току эмиттера через  $RE$ , сравнительно с напряжением  $V_B$  на базе. Когда фактор смещения становится не жестким, изменение тока базы, проходящего через  $R1$ , из-за изменения  $HFE$  вызывает значительные изменения  $V_B$ . Эти изменения сказываются на изменении  $I_E$  (ток эмиттера) и  $I_C$  (ток коллектора). Как с универсальной цепью, вы можете улучшить действие стабилизирующей цепи с любым  $k$ -том, увеличив падение напряжения на  $RE$ . Не забывайте, что эти результаты для номинального диапазона  $HFE$  от 50 до 150 плюс температурные эффекты. Нижний минимум значений  $HFE$  требует жесткого отношения смещения для такого же действия.

Превосходящее действие цепи шунтирующего делителя при отношении смещения более, чем 12, сравнительно со стабилизирующей цепью, это сюрприз. Когда смещение цепи шунтирующего делителя жесткое,  $V_C$  строго влияет на отношение в  $R1$  к  $R2$  раз  $V_{BE}$ . Когда  $V_{BE}$  меняется от температуры,  $V_C$ , а, значит,  $I_C$ , меняется приблизительно в  $R1$  к  $R2$  раз изменения  $V_{BE}$ .

Поскольку ток коллектора ( $I_C$ ) играет ведущую роль в определении  $V_C$ ,  $I_C$  широко варьируется для этих отношений жесткого смещения. Когда отношение теряет жесткость, изменение  $V_{BE}$  с температурой, умноженное на действие делителя напряжения, теряет господствующее положение; и в части улучшения, при отношении выше 12, схема шунтирования делителем стабилизации начинает превосходить то, что дает стабилизирующая цепь. Вы можете подсчитать это действие отрицательной обратной связи (от сопротивления в цепи коллектора через  $R1$ ). Поскольку обычно сопротивление в цепи коллектора много больше, чем резистор в цепи эмиттера, стабильность универсальной цепи поддерживается лучше для теряющего жесткость отношения смещения.

Поскольку цепь шунтирующего делителя более стабильна, чем шунтирующая цепь, рассматривайте последнюю для применений, где допустима меньшая стабильность, чем предлагаемая стабилизацией или универсальной схемой. А, так как сохраняется роль шунтирующего конденсатора эмиттерной цепи в универсальной и стабилизирующей схемах, шунтирующий делитель может быть более эффективен по стоимости. Отрицательная обратная связь через  $R_1$  в цепи шунтирующего делителя уменьшает входное сопротивление и может улучшить полосу частот и уменьшить искажения, а также уменьшить выходное сопротивление таким же образом, как в универсальной цепи. Вновь вы можете убрать эти эффекты со снижающим конденсатором из центра  $R_1$ . Этот конденсатор обычно много меньше, чем шунтирующий эмиттерный для стабилизированной схемы.

Поскольку ток смещения для цепи шунтирования смещения состоит только из тока базы, он имеет отношение порядка  $IC$  к  $IR_1$ , именно  $hFE$ , и изображается как одна точка. Когда отношение смещения для универсальной схемы и шунтирующего делителя увеличивается, значение  $R_2$  увеличивается, пока не становится бесконечным при  $hFE$  порядка 100. При этих условиях отношение смещения цепи сходится к отношению шунтирующей цепи.

Рис. 8.2 может привести к нескольким общим заключениям. Универсальная цепь имеет лучшую стабильность в широком диапазоне отношений смещения. Цепь стабилизации имеет хорошую стабильность для жестких отношений смещения, но вы должны быть внимательны, когда отношение смещения превосходит 12. И, наконец, цепь шунтирующего делителя существенно лучше шунтирующей цепи и лучше, чем стабилизирующая цепь, для больших отношений смещения.

## 8.2 Техники симуляции

Предыдущий раздел рассматривал графический метод, но более общим методом может быть использование симуляции для определения всех возможных вариаций для данной схемы (включая  $hFE$ , температуру, регулирование напряжения и т. д.); так что проблема в том, какой вид обратной связи можно использовать, какой нет. Извините, но здесь нет прямого ответа, поскольку это касается, например, стоимостных последствий, или последствий исполнения<sup>10</sup>.

В любом случае нам нужно оценить различные способы смещения, используя средства симуляции. Один из анализов будет сделан в разделе разработки РА.

---

<sup>10</sup> Эта позиция явно не понятна при обсуждении с маркетинговыми, или разрабатывающими, или исследовательскими группами, кто знает почему?

## 9 VJT моделирование и контроль

### предупреждение

*Эта глава будет посвящена описанию решений при разработке RF с использованием QUCS. Автор подразумевает, что основные манипуляции с qucs известны. Далее вы найдете, в основном, описания для MacOSX, не доступные для linux или unices архитектур.*

### 9.1 Выбор транзистора

Выбор был сделан в пользу Philips RF библиотеки широкополосных транзисторов. Эти компоненты легко найти по приемлемым ценам.

Список можно найти на <http://www.semiconductors.philips.com/>.

Резюме по этим транзисторам можно найти на рис. 9.1.

Я не буду обсуждать в дальнейшем причин<sup>11</sup> окончательного выбора, но BF G425w кандидат. Он предлагает большое усиление с низким уровнем шумов (если рассматривать LNA), высокую граничную частоту ( 25 GHz ), его эмиттер имеет тепловой отвод, низкая емкость обратной связи. Это устройство может быть использовано в RF интерфейсе, аналоговом или цифровом мобильнике, радарном детекторе, пейджерах, SATV, осцилляторах. Он есть в корпусе SOT343R, подходящем для малой интеграции. Максимально достижимое усиление 20 dB с 25 mA,  $V_{ce} = 2$  V на 2 GHz и 25°C. Точка пересечения третьего порядка при этих условиях обычно 22dBm. Эти параметры должны быть совместимы с нашими нуждами. Вот параметры spice этого устройства.

```
.SUBCKT BFG425W 1 2 3
L1 2 5 1.1E-09
L2 1 4 1.1E-09
L3 3 6 0.25E-09
Ccb 4 5 2.0E-15
Cbe 5 6 80.0E-15
Cce 4 6 80.0E-15
Cbpb 5 7 1.45E-13
Cbpc 4 8 1.45E-13
Rsb1 6 7 25
Rsb2 6 8 19
```

---

<sup>11</sup> Из-за тока,  $F_t$ ,  $V_{ce}$ , мощности рассеяния и т. д.

[Go to the interactive version of this Selection Guide.](#)

Go to the interactive version of this Selection Guide.  
Download in MS-Excel sheet (right-click and choose Save As...)

[illegible]

Рис. 9.1. Таблица транзисторов от philips semiconductor

```
Q1 4 5 6 6 NPN
```

```
.MODEL NPN NPN
```

```
+ IS = 4.717E-17 + BF = 145 + NF = 0.9934  
+ VAF = 31.12 + IKF = 0.304 + ISE = 3.002E-13  
+NE =3 +BR =11.37 +NR =0.985  
+ VAR = 1.874 + IKR = 0.121 + ISC = 4.848E-16  
+NC =1.546 +RB =14.41 +IRB =0  
+ RBM = 6.175 + RE = 0.1779 + RC = 1.780  
+ CJE = 3.109E-13 + VJE = 0.9 + MJE = 0.3456  
+ CJC = 1.377E-13 + VJC = 0.5569 + MJC = 0.2079  
+ CJS = 6.675E-13 + VJS = 0.4183 + MJS = 0.2391  
+ XCJC = 0.5 + TR = 0.0 + TF = 4.122E-12  
+ XTF = 68.2 + VTF = 2.004 + ITF = 1.525  
+PTF =0 +FC =0.5501 +EG =1.11  
+XTI =3 +XTB =1.5  
.ENDS
```

Поскольку модель, используемая в SPICE и в QUCS, полагается на gummel-roop моделизацию, и поскольку уровень моделизации одинаков, может использоваться некое прямое преобразование для создания библиотеки в QUCS.

Для прямого использования этого файла вам нужно сохранить его в другой директории, чем проект (небольшая ошибка имеет место при подсчете). Затем он будет работать, но есть еще несколько вопросов по, собственно, параметрам. И есть резон пойти дальше другим путем.

Справочные данные можно найти на web-сайте philips.

## NPN 25 GHz wideband transistor

BFG425W

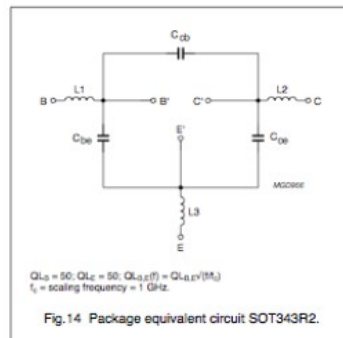
## SPICE parameters for the BFG425W die

SEQUENCE No.	PARAMETER	VALUE	UNIT
1	IS	47.17	αA
2	BF	145.0	—
3	NF	0.993	—
4	VA	31.12	V
5	IKF	304.0	mA
6	ISE	300.2	fA
7	NE	3.000	—
8	BR	11.37	—
9	NR	0.985	—
10	VAR	1.874	V
11	IKR	0.121	A
12	ISC	484.8	αA
13	NC	1.546	—
14	RB	14.41	Ω
15	IRB	0.000	A
16	RBM	6.175	Ω
17	RE	177.9	mΩ
18	RC	1.780	Ω
19 <sup>(1)</sup>	XTB	1.500	—
20 <sup>(1)</sup>	EG	1.110	eV
21 <sup>(1)</sup>	XTI	3.000	—
22	CJE	310.9	fF
23	VJE	900.0	mV
24	MJE	0.346	—
25	TF	4.122	ps
26	XTF	68.20	—
27	VTF	2.004	V
28	ITF	1.525	A
29	PTF	0.000	deg
30	CJC	137.7	fF
31	VJC	556.9	mV
32	MJC	0.207	—
33	XCJC	0.500	—
34 <sup>(1)</sup>	TR	0.000	ns
35 <sup>(1)</sup>	CJS	667.5	fF
36 <sup>(1)</sup>	VJS	418.3	mV
37 <sup>(1)</sup>	MJS	0.239	—
38	FC	0.550	—

SEQUENCE No.	PARAMETER	VALUE	UNIT
39 <sup>(2)(3)</sup>	C <sub>cb</sub>	145	fF
40 <sup>(2)</sup>	R <sub>b21</sub>	25	Ω
41 <sup>(3)</sup>	R <sub>b22</sub>	19	Ω

## Notes

- These parameters have not been extracted, the default values are shown.
- Bonding pad capacity C<sub>cb</sub> in series with substrate resistance R<sub>b21</sub> between B' and E'.
- Bonding pad capacity C<sub>cb</sub> in series with substrate resistance R<sub>b22</sub> between C' and E'.



## List of components (see Fig.14)

DESIGNATION	VALUE	UNIT
C <sub>cb</sub>	80	fF
C <sub>cb</sub>	2	fF
C <sub>cb</sub>	80	fF
L1	1.1	nH
L2	1.1	nH
L3 (note 1)	0.25	nH

## Note

- External emitter inductance to be added separately due to the influence of the printed-circuit board.

Рис. 9.2. Параметры spice извлекаются из справочных данных philips

## 9.2 Создание библиотеки

Помните, что при создании устройства почти всегда обязательным для чтения, чтобы понять, как сделана модель, является техническая документация. Это полезно для понимания ограничений, и того, как можно корректировать некоторые данные, если будет нужно. К большому сожалению значительная часть коммерческого программного обеспечения совершенно в стороне от понимания, какую реальную модель они используют, и ее ограничений; QUCS совершенно исключительна в этом плане, ее полное моделирование объяснено теоретически в специальном техническом документе.

В порядке проведения этого теста нам нужно создать модель нашего компонента. Для выполнения этой задачи вы должны создать файл, который содержит все библиотеки, и этот файл сохранить под /usr/local/share/qucs/library/philips\_RF\_widebande\_npn.lib

Вы можете редактировать этот файл с помощью vi. Вам нужно добавить следующие строки:

```
<Qucs Library 0.0.7 "philips RF wideBand">

<Component BFG425W>
<Description>
RF wideband NPN 25GHz
2V, 25mA, 20dB , 2000MHz
Manufacturer: Philips Inc.
NPN complement: BFG425W

based on spice parameter from philips
```

```

sept 2005 thierry
</Description>
<Model>
<_BJT T_BFG425W_ 1 480 280 8 -26 0 0 "npn" 1 "47.17e-10"
1 "1" 1 "1" 1 "0.304" 1 "0.121" 1 "31.12" 1 "1.874" 0
"300.2e-15" 1 "3" 1 "484.8e-10" 1 "1.546" 1 "145" 1 "11.37"
1 "6.175" 1 "0" 1 "1.78" 1 "0.177.9e-3" 1 "0.14.41" 1 "310.9e-15"
1 "0.900" 1 "0.346" 1 "137.7e-15" 1 "0.5569" 1 "0.207" 1 "0.500"
1 "667.5e-15" 1 "0.4183" 1 "0.239" 1 "0.550" 1 "4.122e-12" 1
"68.2" 1 "2.004" 1 "1.525" 1 "0.0" 1 "26.85" 1 "0.0" 0 "1.0" 0
"1.0" 0 "0.0" 0 "1.0" 0 "1.0" 0 "0.0" 0>
</Model>
</Component>

```

Вы можете заменить 1 на 0, это влияет на опцию видимости; фактически, поместив 1 в начале, вы открываете возможность для пользователя изменить и/или просмотреть параметры, которые будут использованы.

Трюк с предоставлением всего требуемого синтаксиса в том, чтобы вставить NPN в схему, сделать копию устройства. Затем вы должны иметь модель в буфере обмена, и только вставить в файл и добавить описание и разметочные знаки языка. Синтаксис пояснен в подсказке (help) вверху описания файла формата qucs. Затем устройство становится видимо в библиотеке компонент, как показано на рис. 9.3.

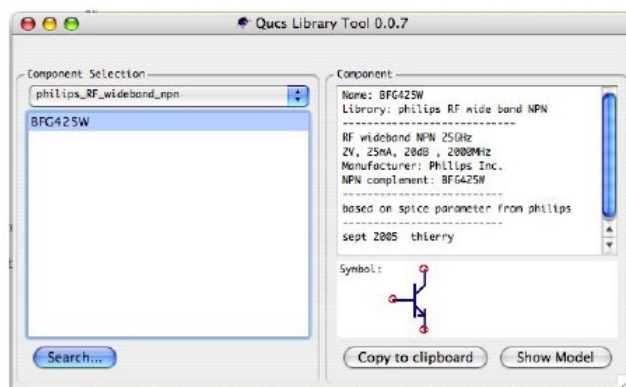


Рис. 9.3. QUCS библиотека компонент с новым компонентом

Сделав это, вы получаете возможность использовать устройство столько раз, сколько вам понадобится, и вы можете отлаживать устройство самым легким образом.

**Предупреждение :** в этом разделе мы опишем только кристалл устройства из-за паразитных элементов корпуса. Мы обязательно опишем схему, но позже.

## 9.3 Проверка библиотеки устройств

Первый шаг перед использованием устройства в приложении – это проверка модели, которую вы используете. Главным образом, поскольку эта модель была создана пользователем. В порядке продолжения, вам нужно обратиться к точным данным: что, к слову, и есть официальные справочные данные.

На этом шаге вам нужно создать проект специально для проверки устройства. Неплохо бы и сохранить путь проверки устройства, поскольку вы будете по-разному его использовать, так что полезна будет возможность вернуться к некоторым симуляциям модели, как таковой. Созданный проект будет выглядеть похожим на тот, что на рис. 9.4.

```

project name : model_verif_bfg425w
project location : $HOME/.qucs/

```

Для законности сделанного мы должны использовать специальное смещение устройства:  $I_c$  должен быть 25mA, следовательно,  $I_b$  должен быть 300  $\mu$  A.

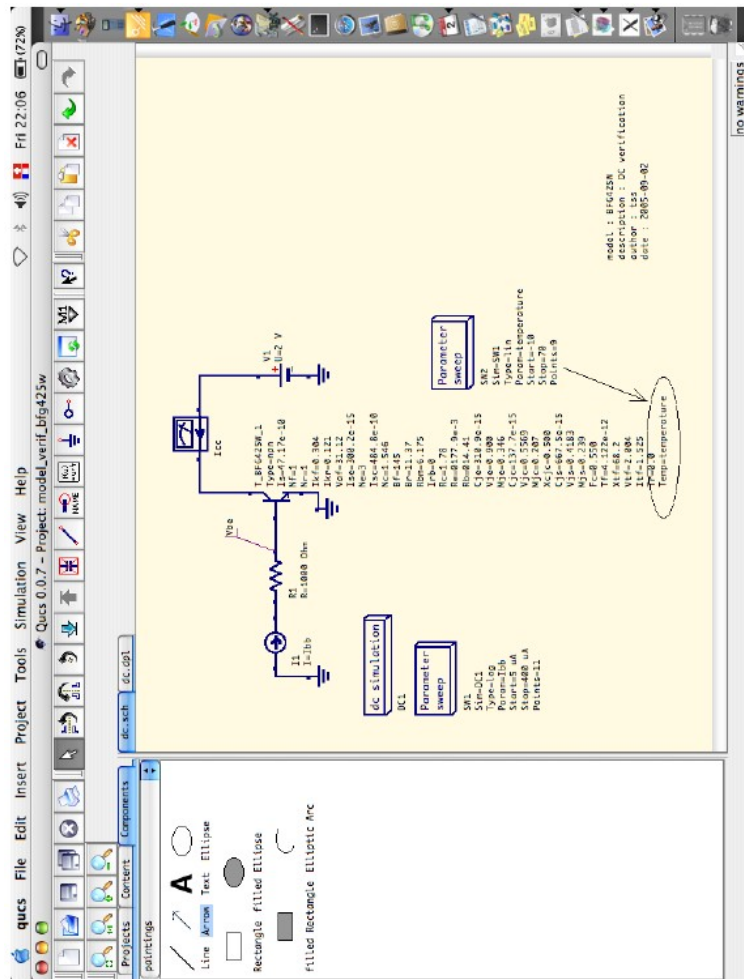


Рис. 9.4. QUCS проект для проверки модели

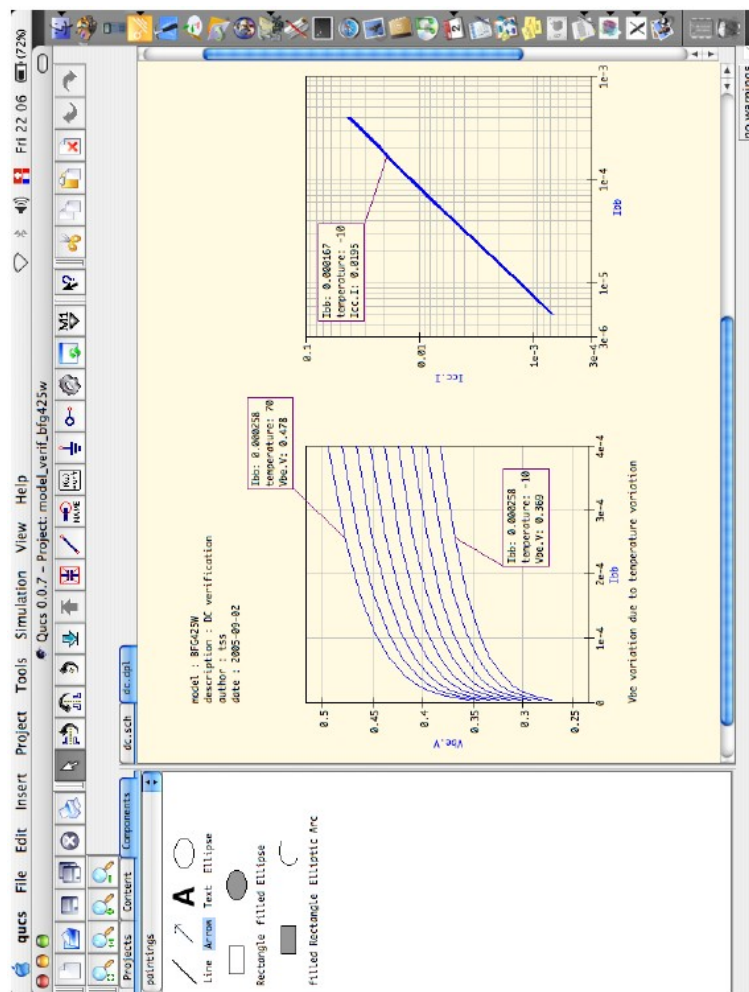


Рис. 9.5. DC проверка и температура

## **9.4 Описание паразитных параметров корпуса**

В порядке правильной симуляции устройства вам нужно использовать корректный корпус, в нашем случае SOT 343R, как упомянуто на web-сайте philips (см. рис. 9.6).

Хотя устройство имеет два эмиттера, модель использует только один эмиттер. Паразитные параметры этой модели показаны в spice netlist, описанном при выборе транзистора и воспроизведенные в схеме (см. рис. 9.8).

Эти параметры всегда критичны при извлечении: либо вы знаете, что с ними делать, либо вы должны довериться той части информации, что дана производителем устройства. Также очень трудно угадать, что должно быть изменено в подобном описании устройства. Некоторая подгонка была выполнена с использованием 3D электромагнитного программного обеспечения в области времен, базирующейся на MOM методах, для проверки этих параметров.

Philips в пятом поколении двух-поликремниевых широкополосных технологий использует крутой легированный эмиттерный профиль, выразившийся в граничной частоте выше 20 GHz, а с поликристаллическими контактами базы было получено низкое ее сопротивление. Через утопленный слой контакт коллектора появляется на верху пластины. Нижний слой соединен прямо с выводом эмиттера корпуса, чем достигается улучшенное температурное поведение (см. рис. 9.7).

Прямо из схемы вы можете редактировать символ, который может быть использован в следующем файле симуляции. Для продолжения нажмите F3 или отредактируйте символ схемы из меню «файл». Просто, нарисовать npn transistor, и возвращайтесь в схему повторным нажатием F3.

Plastic surface mounted package; reverse pinning; 4 leads

SOT343R

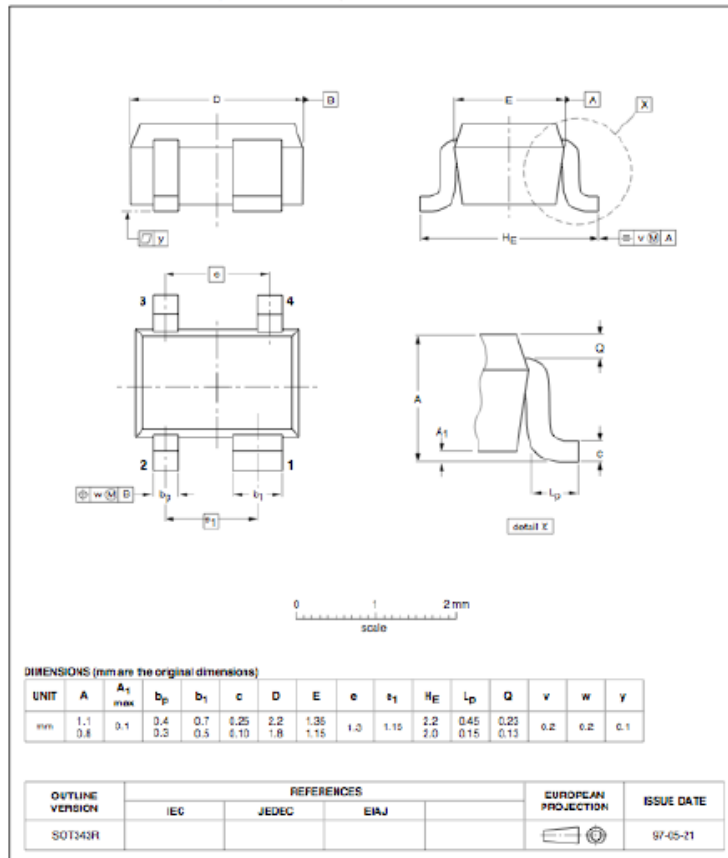


Рис. 9.6. SOT 343R корпус, описание

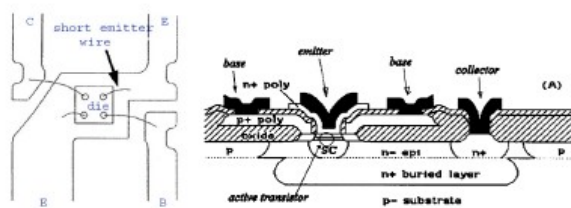


Рис. 9.7. Соединение кристалла в пятом поколении транзисторов от philips

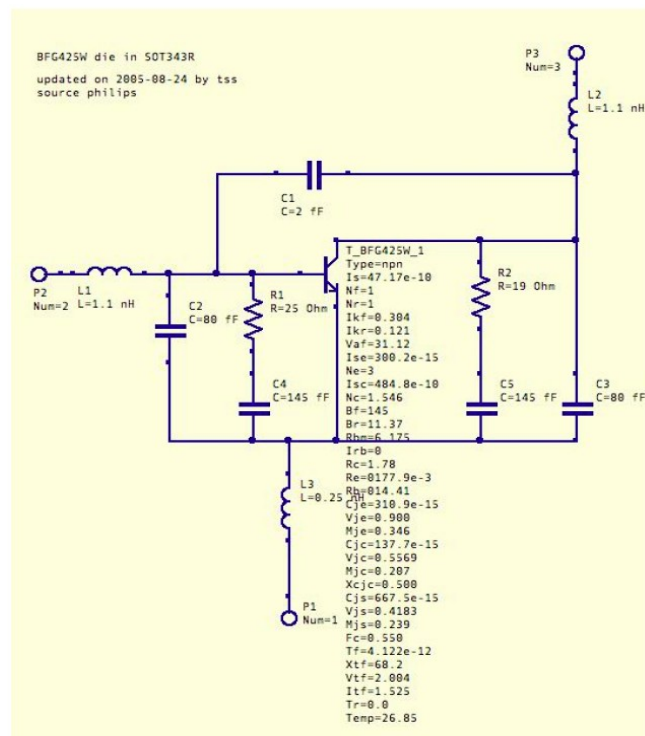


Рис. 9.8. Описание bfg425W в корпусе sot343R

## 9.5 Проверка S-параметров малого сигнала

В этом разделе нам понадобится перерисовать новую схему, используя модель, которую мы создали, плюс поместить некоторые дополнительные компоненты на используемые порты<sup>12</sup>.

Вы должны иметь схему, похожую на ту, что представлена на рис. 9.9.

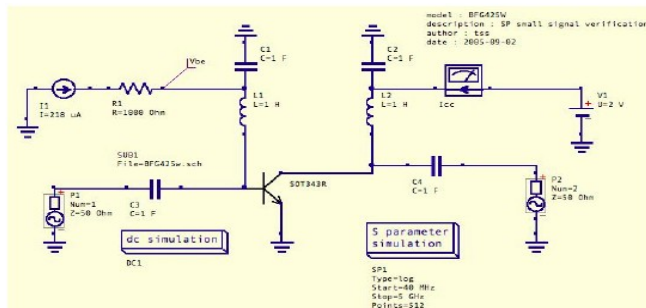
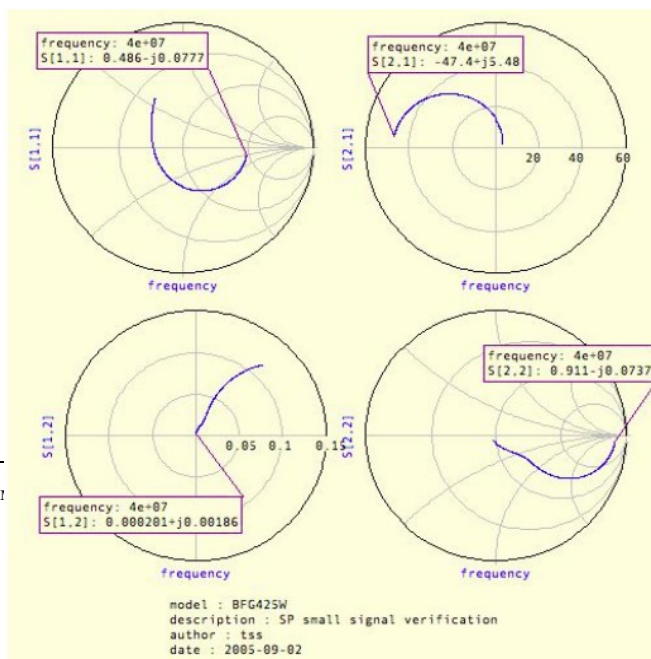


Рис. 9.9. Схема используемая для симуляции S-параметров

Компоненты, использованные для проверки модели должны быть странными (катушка в 1H и конденсатор в 1F). Это нормально, поскольку мы нуждаемся в наличии очень широкой полосы пропускания схемы, и поскольку мы хотим отобразить только активное устройство, и сравнить со справочными данными. Другой путь в использовании DC bloc или DC feed или bias Tee для подключения питания к компоненту. Это правильный путь. Затем вы должны создать дисплей для визуализации S-параметров: обычно s11 и s22 в smith и s12 и s21 в polar. Теперь мы сравним эти результаты с измеренными параметрами от philips:

```
! Filename: 225bfg425.001
! BFG425W Field C1
! V1=8.667E-001V,V2=2.000E+000V, I1=3.585E-004A, I2=2.496E-002A
! S11 S21 S12 S22
!Freq(GHz) Mag Ang Mag Ang Mag Ang Mag Ang
#GHz S MA R50
```

```
0.040 0.325 -8.696 38.472 173.381 0.002 71.865 0.923 -3.072
0.100 0.331 -23.004 37.457 164.549 0.005 83.280 0.915 -9.551
```



<sup>12</sup> Мы применили

те.

Рис. 9.10. S-параметры симуляции для проверки модели

```

0.200 0.315 -44.455 34.771 150.487 0.008 75.947 0.863 -18.965
0.300 0.296 -63.008 31.364 138.811 0.012 71.608 0.794 -26.449
0.400 0.278 -79.654 27.951 128.829 0.015 68.186 0.725 -32.076
0.500 0.265 -94.339 24.856 120.248 0.017 65.974 0.664 -36.332
0.600 0.254 -106.508 22.159 113.362 0.020 64.514 0.613 -39.533
0.700 0.246 -116.820 19.885 107.530 0.022 63.362 0.569 -42.071
0.800 0.240 -126.472 17.964 102.255 0.024 62.701 0.533 -44.121
0.900 0.235 -134.500 16.345 97.645 0.027 61.910 0.504 -45.968
1.000 0.232 -141.743 14.958 93.487 0.029 61.280 0.479 -47.614
1.100 0.230 -148.265 13.770 89.661 0.031 60.570 0.457 -49.172
1.200 0.230 -154.216 12.748 86.091 0.033 59.878 0.438 -50.696
1.300 0.230 -159.761 11.850 82.773 0.036 59.238 0.421 -52.103
1.400 0.231 -164.776 11.070 79.671 0.038 58.509 0.406 -53.483
1.500 0.233 -169.782 10.383 76.687 0.040 57.719 0.392 -54.842
1.600 0.234 -174.382 9.766 73.821 0.043 56.846 0.380 -56.285
1.700 0.236 -178.496 9.213 71.086 0.045 56.001 0.369 -57.740
1.800 0.238 177.334 8.725 68.404 0.047 54.999 0.358 -59.199
1.900 0.241 173.487 8.277 65.836 0.050 53.983 0.348 -60.790
2.000 0.244 169.856 7.874 63.295 0.052 52.923 0.338 -62.399
2.200 0.251 162.836 7.172 58.413 0.057 50.729 0.319 -65.657
2.400 0.259 156.208 6.578 53.682 0.062 48.414 0.301 -68.988
2.600 0.268 150.081 6.068 49.042 0.067 45.958 0.283 -72.558
2.800 0.277 144.221 5.628 44.575 0.072 43.380 0.266 -76.167
3.000 0.288 138.650 5.244 40.174 0.077 40.713 0.248 -80.054
3.500 0.319 125.843 4.470 29.452 0.090 33.634 0.204 -90.648
4.000 0.352 113.999 3.873 18.944 0.102 26.177 0.158 -103.541
4.500 0.389 103.406 3.406 8.713 0.113 18.415 0.113 -121.590
5.000 0.431 92.903 3.011 -1.792 0.123 9.782 0.071 -156.899
5.500 0.463 82.559 2.658 -11.364 0.131 2.534 0.054 148.652
6.000 0.506 73.164 2.374 -21.684 0.138 -6.413 0.095 100.575
6.500 0.516 66.705 2.179 -28.681 0.152 -10.089 0.112 92.309
7.000 0.551 59.664 2.011 -37.894 0.164 -17.920 0.164 82.321
7.500 0.610 50.773 1.808 -49.313 0.166 -29.630 0.246 65.957
8.000 0.644 43.502 1.653 -58.585 0.172 -37.580 0.300 56.971
8.500 0.683 35.816 1.496 -68.478 0.175 -46.984 0.361 47.167
9.000 0.709 27.972 1.338 -77.310 0.173 -55.176 0.412 37.289
9.500 0.736 20.858 1.212 -85.841 0.172 -63.448 0.449 29.117
10.000 0.764 14.187 1.105 -95.600 0.173 -72.751 0.505 22.602
10.500 0.785 7.330 0.997 -104.961 0.171 -81.774 0.554 14.956
11.000 0.802 0.219 0.884 -113.744 0.164 -91.275 0.593 6.422
11.500 0.815 -6.751 0.791 -122.965 0.158 -100.952 0.631 -0.521
12.000 0.822 -13.843 0.690 -131.882 0.149 -111.108 0.667 -8.548

! DEEMBEDDED NOISE DATA
! FREQUENCY FMIN GAMMA OPT Rn
! (GHz) (dB) Mag Ang (NORMALIZED)

```

Используя эти параметры, мы сравним на образце отображения результаты моделирования и измеренные результаты, или прямо покажем ошибки, используя уравнения. Первым делом мы сравним результаты.

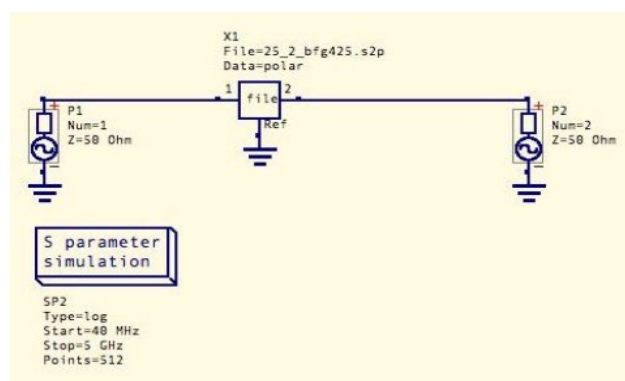


Рис. 9.11. Схема использованная для S-параметров от производителя

На отображение, которое использовано для S-параметров, результат симуляции нашей модели, мы можем добавить результаты из файлов производителя, добавив измерения  $S_{i,j}$  с использованием правого набора данных в диалоге. Вы должны получить разницу между ними.  
Сделав это вы получите результаты, представленные на рис. 9.12.

**ВАЖНОЕ ЗАМЕЧАНИЕ:** Разница, которую вы получите еще исследуется сейчас.

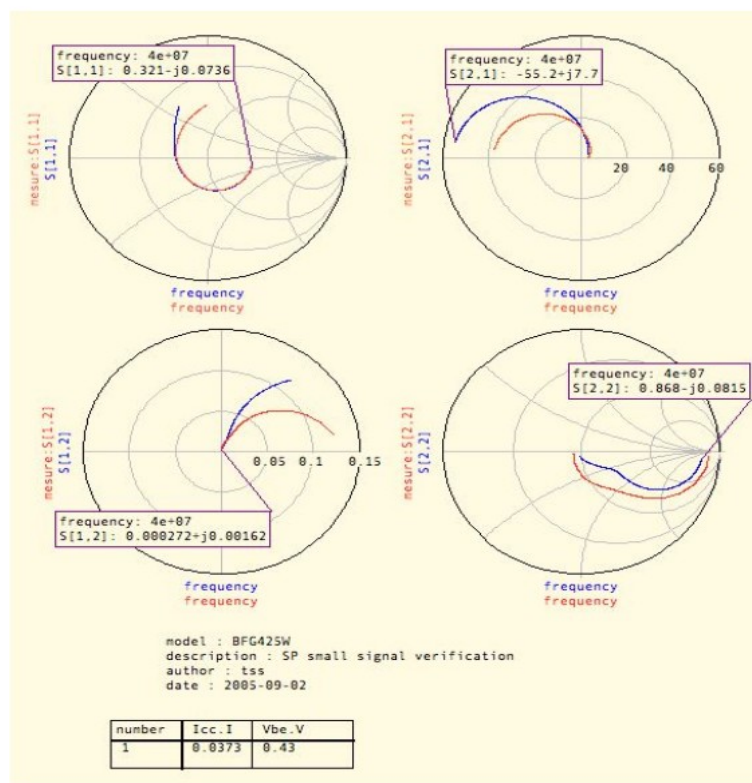


Рис. 9.12. Результаты моделирования и измерений, сравниваемые вместе

# 10 Разработка усилителя мощности

## предостережение

*Эта глава будет посвящена описанию решений при разработке RF с использованием QUCS. Автор подразумевает, что основные манипуляции с qucs известны. Далее вы найдете в основном описания для MacOSX, не доступные для linux или unices архитектур.*

## 10.1 Область интересов

Этот усилитель мощности будет использоваться в более сложной системе, которую я не могу здесь описать, но устройство внутри 868MHz ISM диапазона частот. Этот усилитель рассматривается как усилитель мощности, поскольку он не LNA, но его мощность не столь высока, как это явствует из последующей спецификации системы. Он больше входной усилитель небольшой мощности, питаемый сравнительно малым током. Устройство с действительно большим уровнем мощности, эдак в несколько ватт, будет в другой главе.

## 10.2 Рассмотрение системы

С системной точки зрения нам нужно вначале обозначить, какого рода функции нам нужны. Эти функции будут определены в таблице 10.1.

parameter	description	min	typ	max	unit
$F_o$	frequency of operation	863	868.6	870	MHz
$I_{cc}$	current consumption		20	25	mA
$Z_{in}$	input impedance		50		$\Omega$
$Z_{out}$	output impedance		50		$\Omega$
$P_{in}$	input power	-15	-10	-8	dBm
$P_{out}$	output power	5	10	12	dBm
$V_{cc}$	DC supply voltage	2.45	2.5	2.55	V

Таблица 10.1. Спецификация системы для разработки PA (power amplifier)

Соображения стоимости очень важны, а, значит, можно использовать только один активный компонент, и BOM<sup>13</sup> должен быть сокращен, как только возможно.

Разработка должна работать на FR4 PCB, используемой в производственной линии. Параметры такой основы плохо контролируются, но могут быть описаны, пока вы сохраняете того же поставщика (избегайте странных поставщиков, которые могут изменить состав FR4 без уведомления).

Как отмечено ранее, вы можете описать подложку внутри библиотеки следующими строками:

```
<SUBST FR4_ 100-30 24 00
"4.7" 1 "0.7 mm" 1 "35 um" 1 "2e-4" 1 "0.022e-6" 1 "0.15e-6" 1
>
```

Толщина подложки 0.7mm, но это описывает только один RF слой полной реализации схемы, которая имеет четырехслойную плату. Два внутренних слоя – это питание и земля, верхний и нижний слой – RF.

## 10.3 Соображения по смещению

В этом разделе мы рассмотрим, как сделать смещение, особенно используя эмиттерную обратную связь. Если вы хорошо помните справочные данные транзистора, есть подробное описание  $h_{FE}$  и несколько других элементов, которые следует принять в расчет: сопротивление, напряжение питания, . . .

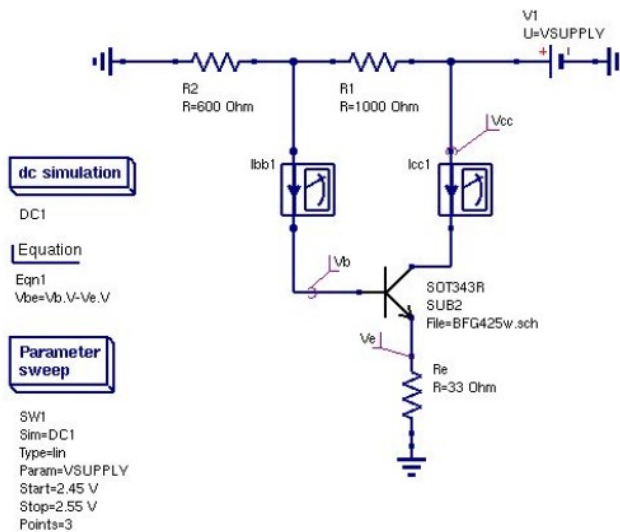


Рис. 10.1. Схема, используемая для изучения

Используемая схема показана на рис. 10.1. Но нам нужно вначале оценить компоненты. Произведя небольшие подсчеты, легко вычислить разные сопротивления: полагая, что

$$I_c = \beta I_b \quad (10.1)$$

$$I_{biasBridge} \ll I_b \quad (10.2)$$

$$I_{biasBridge} = \frac{I_c}{10} \quad (10.3)$$

$$R_e = \frac{V_{cc} - V_{ce}}{I_c} \quad (10.4)$$

$$R_1 + R_2 = \frac{10 \times V_{cc}}{I_c} \quad (10.5)$$

$$R_2 = \frac{10}{I_c} \times (V_{cc} - V_{ce} + V_{be}) \quad (10.6)$$

ВХОДЫ:

- $V_{cc} = 2.5V$
- $V_{be} = 0.412V$
- $I_c = 15mA$

результаты:

- $R_1 = 1K\Omega$
- $R_2 = 600\Omega$
- $R_e = 33\Omega$

Используя эти значения в схеме, мы теперь можем увидеть стабильность разработки. Добавим тот факт, что регулятор напряжения, использованный в данном случае, имеет пульсации в 5 mV в рабочей области. Вам нужно

симулировать DC схему с модификацией BF параметра транзистора от 50 до 120 (поскольку эта возможность не доступна в текущей версии Qucs 0.0.7).

$V_{cc}$ vs $\beta$	50	80	120
2.45	12.21	13.34	14.07
2.50	12.62	13.78	14.54
2.55	13.03	14.23	15.01

Таблица 10.2. Вариации  $I_c$  в mA из-за  $V_{cc}$  и  $\beta$

Из этой таблицы мы можем извлечь некоторый показатель стабильности:

$$\frac{\Delta I_{cc}}{\Delta V}|_{\beta=80} = 8.9 \mu A / mV \quad (10.7)$$

$$\frac{\Delta I_{cc}}{\Delta \beta}|_{V_{cc}=2.5} = 30 \mu A \quad (10.8)$$

$$\frac{\Delta I_{cc}}{\Delta T}|_{\beta=seenote, V_{cc}=2.5} = \dots \mu A / C \quad (10.9)$$

**Заметьте:** для получения температурной зависимости нам нужно взять минимальное  $\beta$  для минимальной температуры, и максимальное  $\beta$  для максимальной температуры.

## 10.4 Почему температурная разработка?

Объективно температурная разработка в электронном оборудовании осуществляется для поддержки при небольшом подъеме температуры,  $\Delta T$ , выше температуры окружающей среды, и как правило при производстве электронных компонент.

На практике, небольшой от  $3^{\circ}\text{C}$  до  $5^{\circ}\text{C}$  подъем температуры компонента почти неизбежен, и в действительности даже желателен. Если подъем меньше этого, могут быть даже большие проблемы, относящиеся к влажности, коррозии и появлению токов утечки.

- Улучшение поведения: отсутствует калибровочный дрейф, поддерживается фазовый сдвиг в контуре обратной связи, стабилизируется усиление, ...
- Улучшение надежности: останавливается механизм, ускоряемый при больших температурах, переноса металла, увеличивается подвижность ионов, ...  
У большинства электронных компонент диапазон отказов удваивается для подъема температуры в  $10^{\circ}\text{C}$  –  $15^{\circ}\text{C}$ , и подъем экспоненциальный! А температурные колебания даже хуже. Подъем температуры практически затруднен для компонент, которые зависят от внутренних жидкостей, таких как электролитические конденсаторы, батареи и подшипники со смазкой.  
Усложненная температурная разработка становится необходима, когда устройства уменьшаются, что увеличивает плотность монтажа. Примеры: VLSIC и технология поверхностного монтажа SMT.
- Улучшение жизни: увеличение  $\Delta T$  увеличивает механический стресс, сказывается на соединениях, металлизации контактов,...

### 10.4.1 Управление температурой

Объективно управление температурой – это разработка внутреннего температурного окружения электронного оборудования так, чтобы исполнение оборудования отвечало ожиданиям покупателя. В диапазоне условий окружающей среды, где оборудование должно эксплуатироваться, оно должно соответствовать спецификации и требуемой надежности. В основном, разработчик слабо контролирует внешнее окружение, так что он должен рассчитывать на расширенный диапазон. Ему больше доступен контроль за внутренним окружением, но его внимание должно быть направлено на конечную цель – поддерживать подходящее окружение для критических компонент. Анализ температурного окружения может обычно делиться на несколько явных частей, определяемых почти изотермическими границами. При рассмотрении типичной закрытой системы изотермические границы можно провести:

- корпус при  $T_e$
- внутреннее пространство при  $T_b$
- компонент при  $T_c$

За счет этих границ,  $\Delta T_{jc}$ ,  $\Delta T_{ca}$  и  $\Delta T_{ja}$  могут быть найдены независимо.  $\Delta T_{ae}$  и  $\Delta T_{e\infty}$  могут также быть найдены независимо для закрытого кожуха, но будут внутренне зависимы для вентилируемого или принудительно охлаждаемого корпуса.

**Аппроксимация проблемы.** На этапе определения продукта выбор корпуса временами диктуется конкурентами, покупателями или маркетинговыми соображениями. Зачастую выбор становится – «как можно меньше», таким образом, невольно переходя на уровень детального рассмотрения, при этом, возможно, следует сделать температурный анализ предлагаемых корпусов. Если окружение, созданное для компонента, не подходит, тогда должен быть разработан дополнительный механизм охлаждения. Одно из приближений – в упрощении проблемы до одномерного анализа. Источники тепла надумано ровно распределены по всему пространству. Поверхность корпуса намеренно становится изотермической. Корпус рассматривается как

великолепный проводник тепла (к сожалению, корпуса все больше и больше делаются из пластика, тепло-изолятора, который осложняет этот образец приближения).

Внешнее окружение рассматривается как стена большой комнаты с поверхностью имеющей излучающую способность  $\varepsilon$ , 1.0 при той же температуре,  $T_\infty$ , что и окружающий воздух, и способностью абсорбировать бесконечное количество тепловой энергии.

Тепло передается через проводимость, излучение, свободную конвекцию, вентиляцию и принудительную конвекцию, что в основном представлено уравнением:

$$Q_t = Q_k + Q_r + Q_c + Q_v + Q_f \quad (10.10)$$

Наиболее трудный компонент, температурное сопротивление  $\theta$ , может варьироваться от простого до очень сложного. К счастью, большинство корпусов для электроники не имеют больше трех путей охлаждения, и во многих случаях третий путь второстепенный, им можно пренебречь для облегчения расчетов. Вот некоторые в основном принимаемые директивы, которые могут использоваться для быстрой оценки конфигурации разработки. Они были получены из примечаний снабженных [?].

Максимальная плотность энергии:

- для маленьких окрашенных равномерно подогреваемых герметизированных корпусов
  - естественно охлаждаемых  $< 4\text{mW/cm}^3$
  - выше  $60\text{cm} < 2\text{mW/cm}^3$
- для естественно охлаждаемых крашенных плат  $< 16\text{mW/cm}^2$
- для крашенных плат с принудительным охлаждением воздуха  $< 110\text{mW/cm}^2$
- для маленьких (60cm или меньше) охлаждаемых форсированной тягой корпусов  $< 20\text{mW/cm}^3$

скорости принудительного обдува:

- для PCB кассет  $> 4\text{m/sec}$
- для корпусов  $< 7.6\text{m/sec}$

## 10.5 Рассеивание DC мощности

Важным вопросом разработки усилителя мощности является рассеивание мощности. Даже, если в этом частном случае, рассеивания мощности нет, что очевидно, не плохо бы посмотреть, как мы смогли бы осуществить это.

Как студент, вы всегда знаете, что вы можете применить закон Кирхгофа к температуре. Вот то небольшое, что вы должны знать – соответствие:

**Температура:** эквивалентна напряжению

**Мощность:** эквивалентна току

**Тепловое сопротивление:** эквивалентно сопротивлению

Вы также должны принимать в расчет некоторую тепловую емкость и возмущения от близких воздействий других источников тепла, в разработке динамики, но здесь мы будем рассматривать только рассеивание мощности DC . . . отталкиваясь от этой точки, вы можете затем представить все, что захотите.

В порядке продолжения, нам нужно создать модель для этого рассеяния мощности. Модель может быть очень

простой в ее понимании, но очень сложной из-за того, что все параметры не достаточно хорошо известны. Следовательно, нам нужно уменьшить уровень моделизации, который будет использован. Вот входные параметры:

- DC рассеиваемая мощность  $15\text{mA} \times 2.5\text{Volts} = 37.5\text{mW}$
- тепловое сопротивление устройства  $\theta_{\text{junction-solder}} = 350\text{degC/W}$
- тепловое сопротивление окружения  $\theta_{\text{h}_{\text{pcb-air}}} = 22\text{degC/W}$
- окружающая температура варьируется от  $-25\text{degC}$  до  $75\text{degC}$ , норма  $25\text{degC}$

Схема, использованная для этой симуляции показана на рис. 10.2<sup>14</sup>.

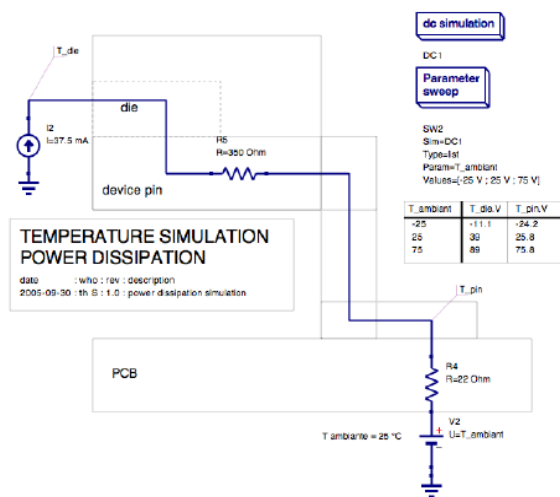


Рис. 10.2. Схема, использованная для симуляции рассеивания мощности DC

## 10.6 Анализ на маленьком сигнале

Текущая версия QUCS не включает обработки гармонического баланса, так что нам нужно сделать некоторые другие симуляции в порядке получения каких-то идей по осуществлению нашей разработки.

<sup>14</sup> Отметьте возможность размещения результатов симуляции непосредственно на схеме, и некоторых компонент, таких как имя документа, исправления и т. д.

# 11 Разработка мал шумящего усилителя

В этом разделе будут описаны два каскада LNA. Основная цель – показать, как мы можем разработать эти LNA, используя программное обеспечение QUCS, но также найти новаторскую разработку для энергосберегающих<sup>15</sup> решений.

Основная разница, как вы должны знать, между PA (усилитель мощности) и LNA (low noise amplifier – мал шумящий усилитель) в том, что в разработке LNA фактор шума решающий, и, следовательно, разработка должна быть несколько иной, чем при разработке только усиления. Это правило разработки хорошо объясняется на всех курсах по RF, так что я сразу перейду к решению, объясняя «пирог», а не «рецепт»!

Как отмечалось ранее, особое внимание будет уделено изучению DC, поскольку полное потребление тока – это критическая точка, а фактор шума, это то, что мы должны получить.

## 11.0.1 Системные соображения

С системной точки зрения нам нужно вначале обозначить, какого рода функции нам нужны. Эти функции будут собраны в таблице 11.1.

parameter	description	min	typ	max	unit
$F_o$	frequency of operation	863	868.6	870	MHz
$I_{cc}$	current consumption		0.5	1	mA
$Z_{in}$	input impedance		50		$\Omega$
$Z_{out}$	output impedance		50		$\Omega$
$P_{in}$	input power	-120	-110	-90	dBm
$V_{cc}$	DC supply voltage	1.4	1.5	1.6	V

Таблица 11.1. Спецификация системы для разработки LNA

**Примечание:** для питающего напряжения DC, мы должны найти корректные пульсации, которые приемлемы в данной разработке в плане возможности специфицировать регулятор напряжения и его PSRR относительно другого напряжения в разработке. Чтобы продолжить, по причине того, что некоторая функциональность еще отсутствует в QUCS<sup>16</sup>, мы будем использовать некоторые обходные пути для изучения DC.

## 11.0.2 Выбор транзистора

В плане разработки LNA особое внимание должно быть уделено этому выбору. Следовательно, вам нужно иметь транзистор, который хорошо разработан для очень маленького тока и применения в LNA.

Я буду использовать BF G403AW от philips<sup>17</sup>. Этот транзистор принадлежит к 5му поколению.

Для непосредственной классификации разных транзисторов, должны быть использованы разные версии.

Параметры следующие:

```
TO BE UPDATED WITH THE CORRECT ONE
.SUBCKT BFG403W 1 2 3
L1 2 5 1.1E-09
L2 1 4 1.1E-09
L3 3 6 0.25E-09
Ccb 4 5 2.0E-15
```

<sup>15</sup> Будьте внимательны, когда я обычно использую термин низкое потребление, я имею ввиду крайне низкое потребление, ниже mA в основном.

<sup>16</sup> Обычно это еще в стадии разработки . . .

<sup>17</sup> У меня нет особой тяги к philips, но они легко обеспечивают некоторыми прототипами и моделями своих транзисторов, более того их стратегия – продолжать поддерживать маленькие широкополосные RF транзисторы, так что, почему бы и нет?

```

Cbe 5 6 80.0E-15
Cce 4 6 80.0E-15
Cbpb 5 7 1.45E-13
Cbpc 4 8 1.45E-13
Rsb1 6 7 25
Rsb2 6 8 19
Q1 4 5 6 6 NPN

.MODEL NPN NPN
bla bla bla
bla bla bla
bla bla bla
bla bla bla
bla bla bla
.ENDS

```

В порядке выполнения некоторых симуляций мы должны добавить этот компонент в библиотеку устройств, как об этом говорилось в главе, посвященной моделированию BJT, и создать схему, использующую это устройство. Паразитные элементы те же, поскольку корпус используется то же, что и для BF G425W.

### 11.0.3 Создание библиотеки

Основной проблемой в этой разработке будет тот факт, что нужный ток для LNA не упоминается в уже измеренных S-параметрах от производителя. Это одна из причин, почему нам нужна специальная нелинейная модель для описания транзистора. Конечно, должен быть сделан предварительный расчет с использованием этих типовых параметров, но, поскольку нам нужны также некоторые другие свойства, такие как искажения и т. д., нелинейная модель то, что нужно.

В порядке ведения этой проверки нам нужно создать модель нашего компонента. Для выполнения этого вы должны создать или отредактировать файл, содержащий все библиотеки. Файл расположен

```
/usr/local/share/qucs/library/philips_RF_widebande_npn.lib
```

Вы можете редактировать этот файл с помощью редактора vi. Вам нужно добавить следующие строки:

```

<Qucs Library 0.0.7 "philips RF wideBand">
...
...
...
<Component BFG403W>
  <Description>
    RF wideband NPN 25GHz
    2V, 3mA, 20dB , 2000MHz
    Manufacturer: Philips Inc.
    NPN complement: BFG403W
    based on spice parameter from philips
    sept 2005 thierry
  </Description>
  <Model>
<_BJT T_BFG403W_ 1 480 280 8 -26 0 0 "npn" bla bla bla bla>
  </Model>
</Component>
...
...
...

```

### 11.0.4 DC изучение

### 11.0.5 SP изучение

### **11.0.6 Изучение нелинейности**

### **11.0.7 Возможные хитрости по улучшению**

## 12 Разработка микрополосковых линий

### 12.1 10dB направленный разветвитель, разработка

Картинки ниже показывают две параллельные проводящие полосы на диэлектрической подложке с металлизацией обратной стороны. Обе проводящие полосы имеют ширину  $W$ , высоту  $t$  и длину  $l$ . Есть конечный зазор  $S$  между полосками. Высота подложки обозначена как  $h$ . С зазором между проводящими полосками обнаруживается достаточно маленькие емкостная и индуктивная связь.

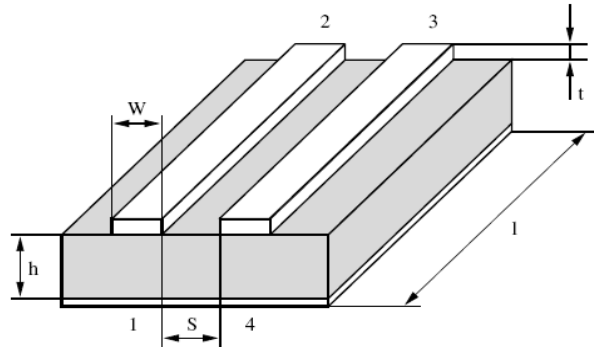


Рис. 12.1. Микрополосковый направленный разветвитель

Такая микрополосковая структура называется «микрополосковые связанные линии». Также определены на рис. 12.1 порты номер 1. . . 4.

#### 12.1.1 Немного предварительной скучной теории

Есть два типа направленных ответвителей: обратный (ответвление от 1 к порту 4) и прямой (ответвление от порта 1 к порту 3) ответвители.

S-параметры идеального направленного обратного ответвителя следующие (с  $C$  обозначением коэффициента связи).

$$\begin{aligned} S_{21} &= \sqrt{1 - C^2} \\ S_{41} &= C \\ S_{31} &= 0 \\ S_{11} = S_{22} = S_{33} = S_{44} &= 0 \end{aligned}$$

В трех проводниковой системе – а именно таковы микрополосковые ответвители – есть два типа режимов: четный и нечетный. Следовательно такая система описывается нечетными и четными характеристическими импедансами ( $Z_{L,o}$  и  $Z_{L,e}$ ) и нечетными и четными эффективными диэлектрическими константами ( $\epsilon_{r,eff,o}$  и  $\epsilon_{r,eff,e}$ ). Характеристические уравнения для идеального обратного ответвителя

$$\begin{aligned} \epsilon_{r,eff,e} &= \epsilon_{r,eff,o} \\ Z_{L,e} &\neq Z_{L,o} \end{aligned}$$

и для идеального прямого ответвителя

$$\begin{aligned}\varepsilon_{r,eff,e} &\neq \varepsilon_{r,eff,o} \\ Z_{L,e} &= Z_{L,o}\end{aligned}$$

S-параметры идеального прямого направленного ответвителя следующие.

$$\begin{aligned}S_{21} &= \sqrt{1 - C^2} \\ S_{31} &= C \\ S_{41} &= 0 \\ S_{11} = S_{22} = S_{33} = S_{44} &= 0\end{aligned}$$

Для обоих идеальных – прямого и обратного – ответвителей коэффициент отражения равен нулю. Порт 1 называется инжекторным (впрыскивающим) портом. Порт 2 – передаточный порт. В обратном ответвителе порт 4 – ответвляющий порт, а порт 3 называется изолированным портом. В прямом ответвителе это другой путь обхода.

*Пожалуйста, заметьте:* Данные S-параметры для прямого и обратного ответвителей справедливы для всех концов сторон каждого порта с импедансом связи  $Z_L$  – обычно 50.

## 12.1.2 Уравнения разработки

В микроволновых лабораториях обратные линейные ответвители распространены больше всего. Базовые уравнения разработки могут быть записаны как

$$\begin{aligned}C &= \frac{Z_{L,e} - Z_{L,o}}{Z_{L,e} + Z_{L,o}} \\ \beta \cdot l &= \frac{\pi}{2} \\ Z_L^2 &= Z_{L,o} \cdot Z_{L,e} \\ Z_{L,e} &= Z_L \cdot \sqrt{\frac{1+C}{1-C}} \\ Z_{L,o} &= Z_L \cdot \sqrt{\frac{1-C}{1+C}}\end{aligned}$$

С

$$\begin{aligned}\beta \cdot l &= \frac{\pi}{2} \\ \leadsto l &= \frac{\pi}{2 \cdot \beta} = \frac{\pi \cdot c}{2 \cdot \omega} = \frac{c}{4 \cdot f} = \frac{\lambda}{4}\end{aligned}$$

длина  $l$  такого ответвителя определяется четвертью длины волны. Оба характеристических импеданса могут быть вычислены через импеданс связи  $Z_L$ , то есть, 50, и коэффициент ветвления  $C$ .

## 12.1.3 Использование уравнений разработки

С предыдущими определениями легко разработать 10dB направленный обратный ответвитель. Мы имеем импеданс связи  $Z_L = 50\Omega$  и коэффициент ветвления  $C$  в dB. Вначале мы линеаризуем коэффициент ветвления.

$$\begin{aligned}C_{dB} &= -10\text{dB} \\ \leadsto C &= 10^{C_{dB}/20} = 10^{-0.5} \approx 0.316\end{aligned}$$

Теперь мы вычислим нечетный и четный импедансы.

$$\begin{aligned}Z_{L,e} &= Z_L \cdot \sqrt{\frac{1+C}{1-C}} \approx 69.4\Omega \\ Z_{L,o} &= Z_L \cdot \sqrt{\frac{1-C}{1+C}} \approx 36.0\Omega\end{aligned}$$

## 12.1.4 Что дальше?

Все это скучная теория, можете подумать вы... С импедансами на руках инженер должен уйти в магические диаграммы и найти физические размеры своего ответвителя. Но теперь здесь есть Qucs.

Все становится легче.

Только выберите Инструменты → Расчет линии в основном меню или нажмите **Ctrl+3** для начала расчета линии передачи.

Затем выберите Связанная микрополосковая линия в окне выбора Тип линии передачи. Появится нечто похожее на показанное на рис. 12.2.

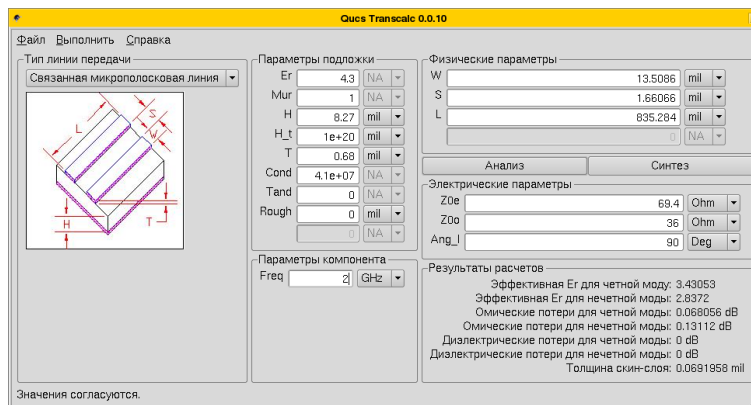


Рис. 12.2. Qucs Transcalc

Введите 69.4 в поле Z0e, 36.0 в поле Z0o и 90 в Ang\_L поле панели Электрические параметры. Поле Ang\_L указывает желаемую электрическую длину линии (помните:  $90^\circ \approx \pi/2$ ). Выберем единицы Deg.

Наша выбранная частота разработки 2GHz. Следовательно, введите это значение в поле Freq панели Параметры компонента.

Затем нажмите клавишу Синтез (Выполнить в меню) или нажмите **F4**. Программа рассчитает физические параметры W, S и L на панели Физические параметры.

*Пожалуйста, заметьте:* В зависимости от подложки (показанной на панели Параметры подложки) вычисленные значения могут варьироваться.

Окончательно мы получаем

$$\begin{aligned} W &= 520 \mu\text{m} \\ S &= 199 \mu\text{m} \\ L &= 14.93\text{mm} \end{aligned}$$

Разработка закончена... Чувствуете себя немного лучше?

## 12.1.5 Проверка разработки

Ок. Давайте проверим, что мы так долго разрабатывали. Выберите Выполнить Скопировать в буфер из меню или нажмите **F2**. Это скопирует текущую, показанную микрополосковую линию в Qucs Transcalc, в общий буфер обмена.

Теперь переключитесь в пустое окно схемы Qucs и нажмите **Ctrl+V**. Этим вы вставите содержимое буфера обмена, и щелкните левой клавишей мышки, чтобы разместить выбранное на схеме. Это должно дать вам нечто похожее на рис. 12.3.

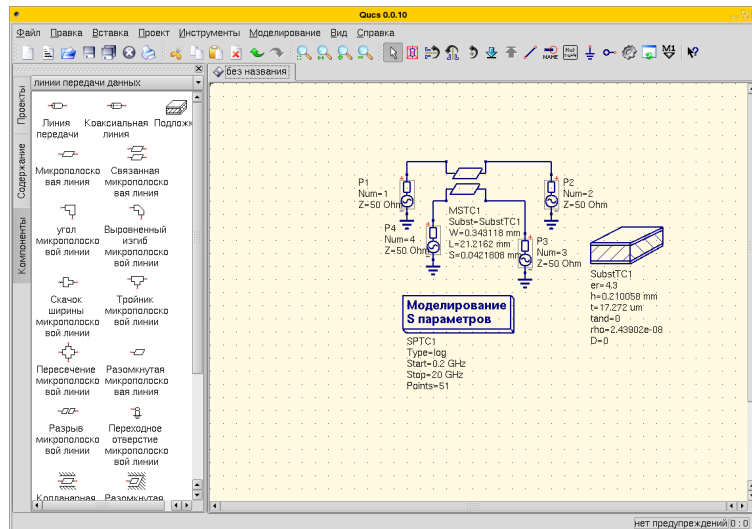


Рис. 12.3. Связанные микрополосковые линии в схеме Qucs

Теперь нажмите клавишу уравнения (показанную на рис. 12.4) на инструментальной панели Qucs.



Рис. 12.4. Клавиша уравнения

Поместите уравнение в схему и введите следующее уравнения. Нажмите Добавить в диалоге уравнений (см. рис. 12.5) для добавления новых уравнений. И наконец нажмите клавишу ОК.

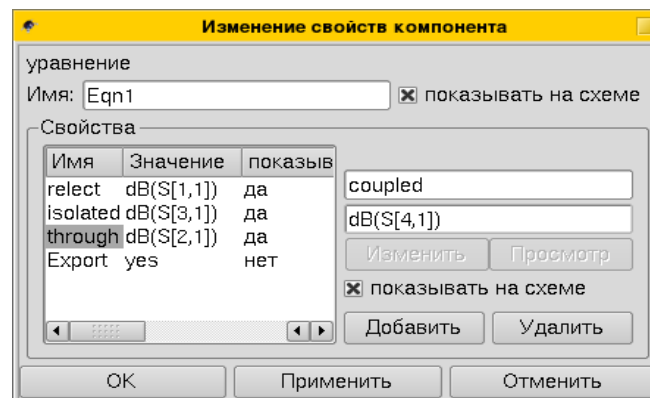


Рис. 12.5. Диалог уравнения

Также отредактируем свойство компонента MSTC1, уменьшив некоторые цифры. Это обеспечит способность нашей технологии использовать эти значения, когда (если) они решат производить вашу разработку. Теперь отредактируем свойства симуляции S-параметров. Вы должны сделать это либо дважды щелкнув по компоненту и используя диалог компонентов, либо вы можете непосредственно щелкнуть по значениям в схеме и вставить 0.2 GHz для Start, 4.2 GHz для Stop и 101 для Points.

В завершение сохраните вашу схему, нажав **Ctrl+S**. Проверьте, все ли выглядит похоже на рис. 12.6.

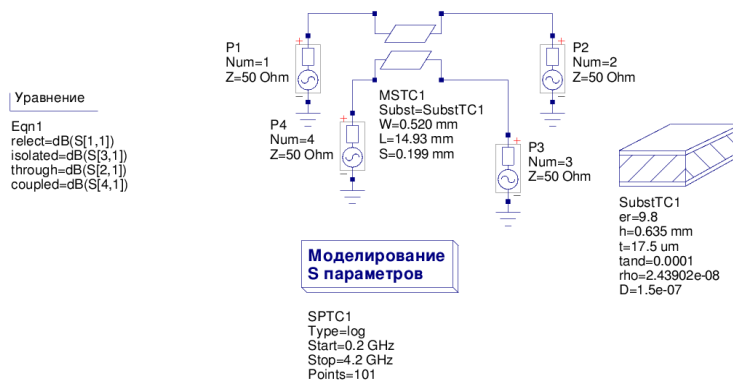


Рис. 12.6. Окончательная схема микрополоскового ответвителя

Теперь выберите **Моделирование** **Моделировать** из основного меню или просто нажмите **F2** для симуляции схемы.

Когда окно симуляции исчезнет, выберите Декартовы координаты (Cartesian diagram) из левого окна выбора и поместите диаграмму в (пока пустое) окно отображения данных. Дважды щелкните по through, reflect, isolated и coupled данным с тем, чтобы добавить их в диаграмму, внутри окна диалога диаграммы, как показано на рис. 12.7.

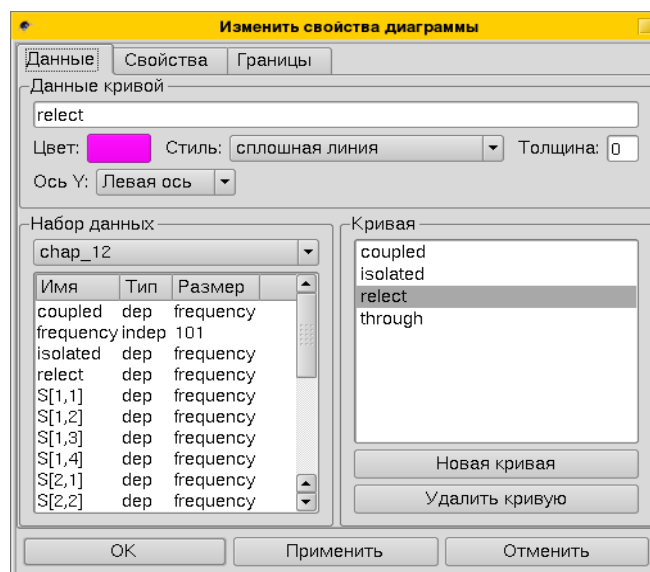


Рис. 12.7. Диалог диаграммы

Нажмите **ОК** для завершения диалога диаграммы. После всего вы увидите следующую диаграмму.

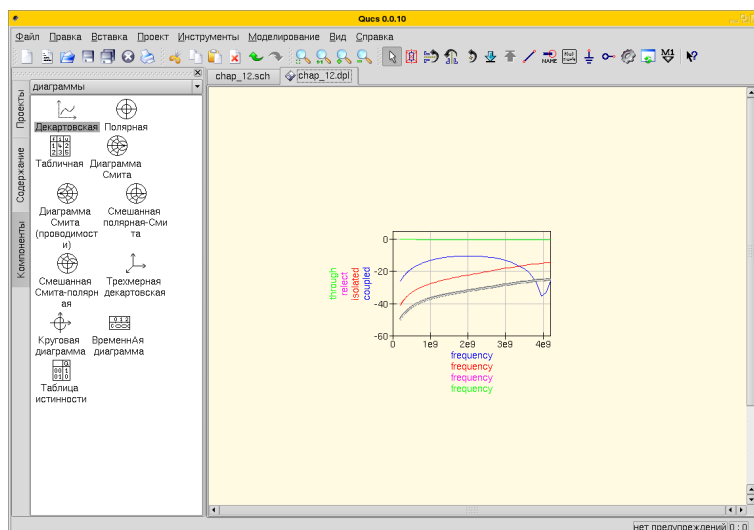


Рис. 12.8. Результаты симуляции микрополоскового ответвителя

## 12.1.6 Предлагаемые улучшения

Используя диалог диаграмм (двойной щелчок по диаграмме), вы можете улучшить<sup>18</sup> визуализацию данных, чтобы видеть ее достойным образом. Я вручную зафиксировал ограничения по y-оси, установил маркеры и установил толщину кривой в 2 точки. Также я ввел общую метку x-оси. Посмотрите на рис. 12.9, как она выглядит теперь.

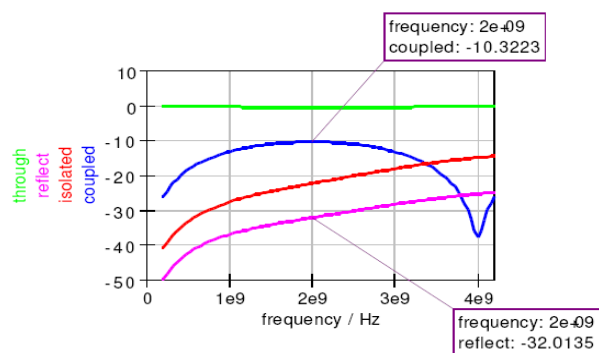


Рис. 12.9. Диаграмма результатов симуляции направленного ответвителя

Маркер на кривой ответвителя показывает коэффициент связи -10.32 на частоте 2GHz (дважды щелкните по маркеру, чтобы изменить точность маркируемых данных). Это размах, которого ради мы пытались осуществить разработку.

Кажется связь между линиями чуть-чуть слаба. Что ж, мы уменьшим зазор между полосковыми проводниками S по 16.5  $\mu$  m до 0.1825 mm, и симулируем опять.

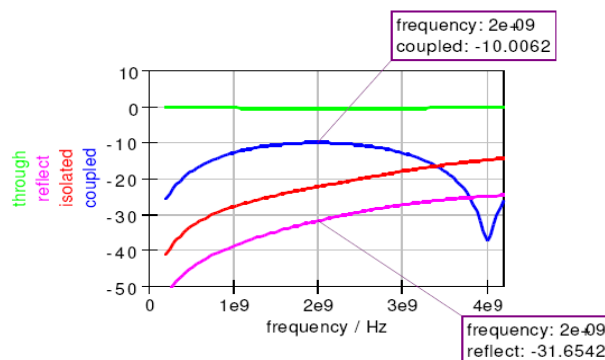


Рис. 12.10. Диаграмма оптимизированных результатов симуляции направленного ответвителя

Окончательно, великолепный<sup>19</sup> 10dB ответвитель показан на рис. 12.10.

### 12.1.7 Оставшиеся размышления

Диаграмма на рис. 12.10 показывает коэффициент отражения около -31.7dB. Изоляция (около -22.2dB) не столь хороша, как планировалась. Так что же случилось с моими уравнениями разработки?

Взглянем на рис. 12.2. В Панели результатов калькуляции вы видите **ErEff Even** и **ErEff Odd** различающиеся значительно, и они не те, что мы ожидали от идеального обратного ответвителя:

$$\mathcal{E}_{r,eff,e} = \mathcal{E}_{r,eff,o}$$

Эта «проблема» происходит от того, что есть два запутывающих диэлектрика: воздух и подложка. Часть электромагнитных полей пересекает воздух и часть из них подложку. Вы можете воспрепятствовать этому диэлектрическим покрытием. Это дороже для производства, но улучшит ваши результаты.

## 13 Ссылочное руководство по выражениям измерений

### 13.1 Введение

Данное руководство описывает выражения измерений доступные в «Qucs», «Quite Universal Circuit Simulator – Почти универсальный симулятор схем».

Выражения измерений начинают действовать при запуске пост-процесса результатами симуляции «Qucs», переводя, например, графические выходы напряжения симуляции из вольт в децибелы, или среднеквадратичные значения, или определяя пиковые напряжения. Функции измерений «Qucs» предлагают богатый набор средств манипуляции с данными.

Если вы не вполне знакомы с тем, как вводить эти формулы, пожалуйста, обратитесь к главе «Использование выражений измерений», которая рассказывает о возможностях по созданию и изменению выражений измерения. Там же есть спецификация поддерживаемых типов данных. Глава «Синтаксис функций и обзор» вводит в базовый синтаксис функций и список по категориям всех доступных функций. Ядро документа, детальная компиляция всех функций «Qucs», разделенных по категориям, находится в главах «Математические функции» и «Электрические функции». И, наконец, приложение содержит алфавитный список всех функций.

## 13.2 Использование выражений измерений

Глава описывает использование математических выражений для пост-процесса симуляции данных в «Qucs», как ввести формулы и модифицировать их. Она дает беглое описание всего синтаксиса этих выражений.

### 13.2.1 Ввод выражений измерений

Выражения измерений генерируют новые наборы данных с помощью функций или операторов, ведомых вычислениями результатов симуляции. Эти новые наборы доступны в таблицах отображения данных после симуляции. Соответствующие уравнения могут быть введены в редакторе схем следующими способами:

- Используя иконку уравнений на инструментальной панели (см. рис. 13.1)
- Используя пункт меню «Вставить» → «Вставить уравнение»

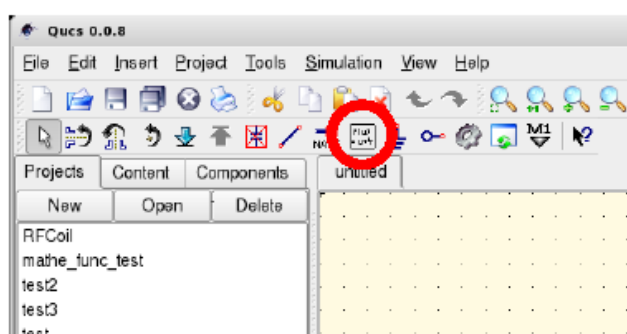


Рис. 13.1. Введение нового выражения измерения с помощью иконки уравнения

Теперь вы можете поместить символ уравнения щелчком мышки где-нибудь на схеме. Каждый щелчок мышки создает новый образец уравнения, содержащий некоторое количество переменных выражения измерения.

Нажмите клавишу **Esc**, если вам больше не нужно вставлять уравнения.

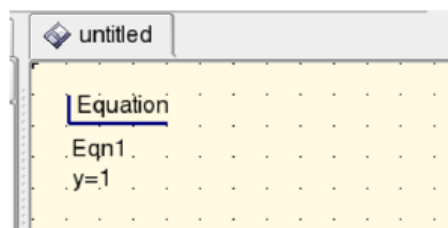
Другая возможность – выбрать существующее уравнение, скопировать его (либо с помощью пункта меню «Правка Копировать», либо с помощью клавиш **Ctrl+C**<sup>20</sup>) и вставить (либо с помощью пункта меню «Правка» → «Вставить» или с помощью **Ctrl + V**).

После успешного создания образца уравнения вы можете модифицировать его.

### 13.2.2 Изменение выражений измерений

Ради простоты мы подразумеваем, что вы только что сгенерировали новое уравнение (если вы изменили существующее, более сложное уравнение, последующие шаги остаются теми же самыми).

Значит, отображение вашей схемы будет выглядеть похожим на то, что на рис. 13.2.



---

<sup>20</sup> **Ctrl + C** означает, что вы должны нажать клавиши **Ctrl** и **C** одновременно.

Рис. 13.2. Вновь созданное уравнение

Теперь вы можете манипулировать текущим именем образца уравнения. Просто щелкните по «Eqn1», которое станет подсвечено, затем введите новое имя для него, и завершите ваши действия с помощью клавиши **Enter**.

После этого вы можете ввести новое уравнение. Вновь щелкните по «у=1». Только «1» будет отмечена, и здесь вы можете ввести новое выражение. Пожалуйста, используйте переменные, операторы и константы, описанные в главе «Синтаксис выражений измерений». Заметьте, что вы можете также ссылаться на результаты (зависимые) других уравнений. Но как изменить имя текущей зависимой «у»? Щелкните правой клавишей мышки по уравнению, и откроется меню содержания. Выберите первый раздел, названный «Редактировать свойства». Откроется новое окно, которое будет выглядеть как на рис. 13.3. Альтернатива для ввода уравнения – двойной щелчок по уравнению.

Теперь вы можете изменить имя зависимой переменной, и само уравнение (которое было «1» в показанном примере), и имя уравнения. Если вы не хотите, чтобы результаты были экспортированы в таблицу отображаемых данных, а только временно нуждаетесь в них для дальнейших вычислений, выберите «нет» в ячейке «Экспорт значения».

### 13.2.3 Синтаксис выражений измерений

Имена функций, имена переменных и имена констант – все они чувствительны к регистру в выражениях измерений. Это выражается в различии между прописными буквами, как 'а' и заглавными, как 'А'.

В функциях запятые используются для разделения аргументов.

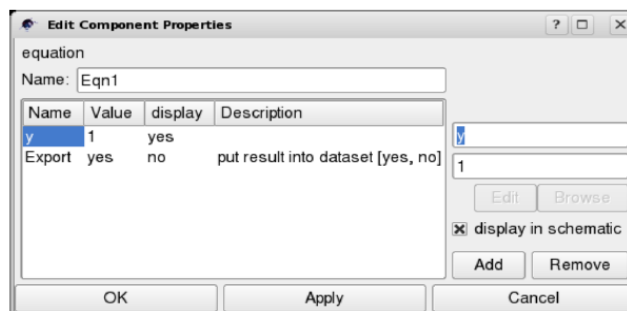


Рис. 13.3. Редактирование свойств уравнения

#### Имена переменных

Определенные пользователем имена переменных состоят из букв, сопровождаемых любым количеством букв, цифр или подчеркиваний.

Синтаксис имен переменных, создаваемых симулятором «Qucs», специфицирован в таблице 13.1. Пожалуйста, отметьте, что все напряжения и токи в «Qucs» – это пиковые значения, исключая напряжения шумов и токов, которые имеют значения rms в полосе 1Hz.

#### Числа

Числа записываются в общепринятой десятичной манере с дополнительной десятичной точкой между цифрами. Для степеней десяти используется хорошо известная научная нотация с 'e'. Таки образом, '1.234e6' – пример для действительного числа с плавающей точкой 1234000. Мнимые числа могут быть введены с множителем 'i' или 'j' (см. также таблицу 13.2). Например, '1+2\*i' или, если вы хотите выбросить знак умножения, '1+i2'.

Встроенные константы

Константы, которые могут использоваться внутри выражений измерений, даны в таблице 13.2.

Variable Name	Description
<i>nodename.V</i>	DC voltage at node <i>nodename</i>
<i>name.I</i>	DC current through circuit component <i>name</i>
<i>nodename.v</i>	AC voltage at node <i>nodename</i>
<i>name.i</i>	AC current through circuit component <i>name</i>
<i>nodename.vn</i>	AC noise voltage at node <i>nodename</i>
<i>name.in</i>	AC noise current through circuit component <i>name</i>
<i>nodename.Vt</i>	Transient voltage at node <i>nodename</i>
<i>name.It</i>	Transient current through circuit component <i>name</i>
<i>name.OP</i>	<i>name</i> = component name, OP = operating point (device dependent), e.g. D1.Id
<i>S[x,y]</i>	S-parameter, e.g. S[1,1]
<i>Rn</i>	equivalent noise resistance
<i>Sopt</i>	optimal reflection coefficient for minimum noise
<i>Fmin</i>	minimum noise figure
<i>F</i>	noise figure
<i>nodename.Vb</i>	Harmonic balance voltage at node <i>nodename</i>

Таблица 13.1. Синтаксис имен переменных, генерируемых симулятором

Constant	Description	Value
e	Euler's constant	2.718282
i , j	Imaginary unit ( $\sqrt{-1}$ )	i1
kB	Boltzmann's constant	1.380658e23 J/K
pi	$\pi$	3.141593

Таблица 13.2. Встроенные константы

Операторы

**Оператор предшествования** Выражения вычисляются стандартным образом, именно – слева направо, пока это не меняется скобками. Приоритет операторов также поддержан общепринятым способом, так, например, умножение имеет преимущество перед сложением. Таблица 13.3 специфицирует в рассортированном виде все операторы так, что те, что выше, имеют более высокий приоритет. Операторы, размещенные на одной линии имеют одинаковый приоритет.

**Диапазоны** Основная номенклатура диапазонов отображена в таблице 13.4. Она показывает одно-мерные диапазоны, хотя и n-мерные диапазоны допустимы, если вы рассматриваете вложенные пространства.

Operator	Name	Example
()	Parentheses, function call	max(v)
^	Exponentiation	3^4
*	Multiplication	3*4
/	Division	3/4
%	Modulo	4%3
+	Addition	3+4
-	Subtraction	3-4
:	Range operator	3:12

Таблица 13.3. Приоритет оператора

Syntax	Explanation
$m:n$	Range from index $m$ to index $n$
$:n$	Range up to index $n$
$m:$	Range starting from index $m$
$:$	No range limitations

Таблица 13.4. Определение диапазонов

### Пост-процесс симуляции данных с выражениями

После запуска симуляции результаты сохраняются в наборе данных. Обычно такие наборы данных – это векторы или матрицы, но могут быть также и действительным или комплексным скаляром. Для анализа переходных процессов эти наборы данных содержат информацию о напряжении или токе по времени, для гармонического анализа они содержат амплитуды и соответствующие частоты, при анализе S-параметров возвращается вектор матриц (то есть, матрицы в зависимости от частоты). В дальнейшем обобщении компонент векторов и матриц содержатся комплексные числа.

Дополнительно наборы данных (datasets) могут генерироваться выражениями. В качестве примера можно указать на функцию `linspace()`, которая создает вектор линейно-протяженных элементов.

## 13.3 Синтаксис функций и обзор

Эта глава представляет введение а базовый синтаксис описания функций и содержит список, упорядоченный по категориям, всех доступных функций.

### 13.3.1 Формат ссылочных функций

«Qucs» предоставляет богатый выбор функций, которые могут использоваться для генерации и отображения наборов данных, вычисляемых на основе функций из результатов симуляции. Помимо большого числа стандартных математических функций, таких как квадратный корень (sqrt), экспоненциальная функция (exp), абсолютное значение (abs), встроены функции особенно полезные для вычислений и преобразований электрических значений. Примером последних служат преобразование Ватт в dBm, генерация шумовых циклов в разработку усилителя, или преобразование S-параметров в Y-параметры.

#### Формат ссылочных функций

В последующих двух главах каждая функция описывается с использованием следующей структуры:

**<Функция Имя>**

Краткое описание функциональности функции.

#### Синтаксис

Определяет основной синтаксис этой функции.

#### Аргументы

Имя, тип, определение диапазона (область определения) и будет ли аргумент необязательным, все будет собрано здесь в таблицу. В случае необязательности параметра специфицируется значение по умолчанию. «Тип - Type» – список, определяющий аргументы, которые допускают и могут содержать следующие символы:

Symbol	Description
$\mathbb{R}$	Real number
$\mathbb{C}$	Complex number
$\mathbb{R}^n$	Vector consisting of $n$ real elements
$\mathbb{C}^n$	Vector consisting of $n$ complex elements
$\mathbb{R}^{m \times n}$	Real matrix consisting of $m$ rows and $n$ columns
$\mathbb{C}^{m \times n}$	Complex matrix consisting of $m$ rows and $n$ columns
$\mathbb{R}^{m \times n \times p}$	Vector of $p$ real $m \times n$ matrices
$\mathbb{C}^{m \times n \times p}$	Vector of $p$ complex $m \times n$ matrices

«Определение диапазона» специфицирует допустимый диапазон. Каждый диапазон представлен в скобках, либо “[”или “]”, означающих, что далее, следующее значение диапазона – это либо включение, либо исключение. Начальное значение отделено от конечного запятой. Затем следует завершающее значение, вновь заканчивающееся скобками, либо “[”или “]”. Первая скобка означает “исключить конечное значение”, вторая означает “включить”.

Если диапазон задается для комплексного числа, оно задается действительным или мнимым значением числа. Если диапазон задан для действительного или комплексного вектора или матрицы, он задает реальное или мнимое значение каждого элемента этого вектора или матрицы. Символы означают «включая перечисленные значения» и «исключая перечисленные значения».

## Описание

Дает более детальное описание того, что делает функция и что она возвращает. В некоторых случаях присутствуют фоновые данные.

## Примеры

Показано применение функции на одном или нескольких простых примерах.

## См. также

Даны ссылки на родственные функции. Щелчок мышки по выбранной ссылке приводит к немедленному переходу к этой функции (только в оригинальной версии).

# 13.3.2 Функции перечисленные по категориям

Эта подборка показывает все «Qucs» функции, отсортированные по категориям. Пожалуйста щелкните по выбранной функции, чтобы перейти к ее описанию (только в оригинальной версии).

## Математические функции

### Векторы и матрицы: создание

`eye()` ... Создает  $n \times n$  единичную матрицу  
`linspace()` ... Создает действительный вектор с линейно протяженными компонентами  
`logspace()` ... Создает действительный вектор с логарифмически протяженными компонентами

### Векторы и матрицы: базовые матричные функции

`adjoint()` ... Сопряженная матрица  
`array()` ... Вывести единичные элементы  
`det()` ... Определитель матрицы  
`inverse()` ... Инверсия матрицы  
`transpose()` ... Транспозиция матрицы

### Элементарные математические функции: базовые действительные и комплексные функции

`abs()` ... Абсолютное значение  
`angle()` ... Фазовый угол в радианах комплексного числа. Синоним для "arg"  
`arg()` ... Фазовый угол в радианах комплексного числа  
`conj()` ... Сопряжение комплексного числа  
`deg2rad()` ... Преобразование фазы из градусов в радианы  
`imag()` ... Мнимое значение комплексного числа  
`mag()` ... Модуль комплексного числа  
`norm()` ... Квадрат абсолютного значения вектора  
`phase()` ... Фазовый угол в градусах комплексного числа  
`polar()` ... Перевод полярных координат в комплексное число  
`rad2deg()` ... Преобразование фазы из градусов в радианы  
`real()` ... Действительное значение комплексного числа  
`signum()` ... Сигнум-функция  
`sign()` ... Знаковая функция  
`sqr()` ... Квадрат числа  
`sqrt()` ... Корень квадратный  
`unwrap()` ... Развертка фазы вектора в радианах

### Элементарные математические функции: экспоненциальная и логарифмическая функции

`exp()` ... Экспоненциальная функция  
`log10()` ... Десятичный логарифм

log2() ... Двоичный логарифм  
ln() ... Натуральный логарифм (по основанию e)

### Элементарные математические функции: тригонометрия

cos() ... Функция косинуса  
cosec() ... Косеканс  
cot() ... Функция котангенса  
sec() ... Секанс  
sin() ... Функция синуса  
tan() ... Функция тангенса

### Элементарные математические функции: Обратные тригонометрические функции

arccos() ... Арккосинус (или «обратный косинус»)  
arccot() ... Арккотангенс  
arcsin() ... Арксинус (или «обратный синус»)  
arctan() ... Арктангенс (или «обратный тангенс»)

### Элементарные математические функции: гиперболические функции

cosh() ... Гиперболический косинус  
cosech() ... Гиперболический косеканс  
coth() ... Гиперболический котангенс  
sech() ... Гиперболический секанс  
sinh() ... Гиперболический синус  
tanh() ... Гиперболический тангенс

### Элементарные математические функции: обратные гиперболические функции

arcosh() ... Гиперболический арка-косинус  
arcoth() ... Гиперболический арка-котангенс  
arsinh() ... Гиперболический арка-синус  
artanh() ... Гиперболический арка-тангенс

### Элементарные математические функции: округление

ceil() ... Округление до следующего большего целого  
fix() ... Усечение десятичного разряда из действительного числа  
floor() ... Округление до следующего меньшего целого  
round() ... Округление до ближайшего целого

### Элементарные математические функции: специальные математические функции

besseli0() ... Модифицированная функция Бесселя порядка ноль  
besselj() ... Функция Бесселя n-го порядка  
bessely() ... Функция Бесселя второго рода n-го порядка  
erf() ... Функция ошибки  
erfc() ... Комплементарная функции ошибки  
erfinv() ... Инверсная функция ошибки  
erfcinv() ... Инверсная комплементарная функция ошибки  
sinc() ... Функция Sinc (синхронизация)  
step() ... Функция Step (шаг)

### Анализ данных: базовые статистики

avg() ... Среднее векторных элементов  
cumavg() ... Совокупное среднее векторных элементов  
max() ... Максимальное значение  
min() ... Минимальное значение  
rms() ... Корень из квадрата среднего векторных элементов  
runavg() ... Скользящее среднее векторных элементов  
stddev() ... Стандартное отклонение векторных элементов  
variance() ... Расхождение векторных элементов

### Анализ данных: базовые операции

cumprod() ... Совокупное произведение векторных элементов  
cumsum() ... Совокупная сумма векторных элементов  
interpolate() ... Эквидистантная сплайн интерполяция данных вектора  
prod() ... Произведение векторных элементов  
sum() ... Сумма векторных элементов  
xvalue() ... Возвращает x-значение, ассоциированное с y-значением  
около a,  
                                специфированного y-значением в заданном векторе  
yvalue() ... Возвращает y-значение заданного вектора,  
локализованного ближе  
                                всего к специфированному x-значению

## Анализ данных: дифференцирование и интегрирование

diff() ... Дифференцировать вектор, касающийся другого вектора  
integrate() ... Интегрировать вектор

## Анализ данных: обработка сигналов

dft() ... Дискретное Фурье преобразование  
fft() ... Быстрое Фурье преобразование  
idft() ... Инверсное дискретное Фурье преобразование  
ifft() ... Инверсное быстрое Фурье преобразование  
Time2Freq() ... Интерпретированное дискретное Фурье преобразование  
Freq2Time() ... Интерпретированное инверсное дискретное Фурье  
преобразование  
kbd() ... Кайзер-Бесселя производное окно

## Электронные функции

### Конверсия единиц

dB() ... dB значение  
dbm() ... Преобразование напряжения в мощность в dBm  
dbm2w() ... Преобразование мощности в dBm в мощность в Watts  
w2dbm() ... Преобразование мощности в Watts в мощность в dBm

### Коэффициенты отражения и VSWR

rtoswr() ... Преобразование k-та отражения в напряжение k-та стоячей  
волны -  
                                voltage standing wave ratio (VSWR)  
rtoz() ... Преобразование k-та отражения в проводимость  
rtoz() ... Преобразование k-та отражения в импеданс  
ytor() ... Преобразование проводимости в k-т отражения  
ztor() ... Преобразование импеданса в k-т отражения

## N-порт матричные преобразования

`stos()` ... Преобразование матрицы S-параметров в матрицу S-параметров с  
разным опорным импедансом(ми)  
`stoy()` ... Преобразование матрицы S-параметров в матрицу Y-параметров  
`stoz()` ... Преобразование матрицы S-параметров в матрицу Z-параметров  
`twoport()` ... Преобразование двух-портовой матрицы из одного представления в  
другое  
`ytos()` ... Преобразование матрицы Y-параметров в матрицу S-параметров  
`ytoz()` ... Преобразование матрицы Y-параметров в матрицу Z-параметров  
`ztos()` ... Преобразование матрицы Z-параметров в матрицу S-параметров  
`ztoy()` ... Преобразование матрицы Z-параметров в матрицу Y-параметров

## Усилители

`GaCircle()` ... Окружности с постоянным допустимым усилением мощности  
Ga в плане источника  
`GpCircle()` ... Окружность(и) с постоянным оперативным усилением  
мощности  
Gp в плане нагрузки  
`Mu()` ... Mu фактор стабильности двух-портовой матрицы S-параметров  
`Mu2()` ... Mu' фактор стабильности двух-портовой матрицы S-параметров  
`NoiseCircle()` ... Генерирует окружность(и) с постоянной фигурой(ми)  
шумов  
`PlotVs()` ... Возвращает группу данных базируемых на векторе или  
матрице  
векторов с зависимостью от данного вектора  
`Rollet()` ... Rollet фактор стабильности двух-портовой матрицы S-параметров  
`StabCircleL()` ... Окружность стабильности в плане нагрузки  
`StabCircleS()` ... Окружность стабильности в плане источника

## 13.4 Математические функции

### 13.4.1 Векторы и матрицы

#### Создание

#### eye()

Создает n x n единичную матрицу.

#### Синтаксис

`y=eye(n)`

#### Аргументы

Name	Type	Def. Range	Required
n	N	$[1, +\infty[$	✓

#### Описание

Эта функция создает n x n единичную матрицу вида

$$\begin{pmatrix} 1 & 0 & \dots & 0 & 0 \\ 0 & 1 & 0 & \dots & 0 \\ \vdots & 0 & \ddots & 0 & \vdots \\ 0 & \dots & 0 & 1 & 0 \\ 0 & 0 & \dots & 0 & 1 \end{pmatrix}$$

#### Пример

`y=eye(2)` возвращает

1	0
0	1

#### См. также

## linspace()

Создает действительный вектор с линейно протяженными компонентами.

### Синтаксис

```
y=linspace(xs,xe,n)
```

### Аргументы

Name	Type	Def. Range	Required
xs	$\mathbf{R}$	$] -\infty, +\infty[$	✓
xe	$\mathbf{R}$	$] -\infty, +\infty[$	✓
n	$\mathbf{N}$	$[2, +\infty[$	✓

### Описание

Эта функция создает действительный вектор с n линейно-протяженными компонентами. Первый компонент – это xs, последний xe.

### Пример

```
y=linspace(1,2,3) В о з в р а щ а е т 1, 1.5, 2.
```

### См. также

```
logspace( )
```

# logspace()

Создает действительный вектор с логарифмически-протяженными элементами.

## Синтаксис

```
y=logspace(xs,xe,n)
```

## Аргументы

Name	Type	Def. Range	Required
xs	<b>R</b>	$] - \infty, + \infty [$	✓
xe	<b>R</b>	$] - \infty, + \infty [$	✓
n	<b>N</b>	$[2, + \infty [$	✓

## Описание

Эта функция создает действительный вектор с n логарифмически-протяженными компонентами. Первый компонент – это xs, последний xe.

## Пример

```
y=logspace(1,2,3) В о з в р а щ а е т 1, 1.41, 2.
```

См. также

```
linspace()
```

Базовые матричные функции

adjoint()

Сопряженная матрица.

Синтаксис

```
Y=adjoint(X)
```

Аргументы

Name	Type	Def. Range	Required
X	$\mathbb{R}^{m \times n}, \mathbb{C}^{m \times n}, \mathbb{R}^{m \times n \times p}, \mathbb{C}^{m \times n \times p}$	$] - \infty, + \infty[$	✓

Описание

Эта функция вычисляет сопряженную матрицу Y к матрице X :

$Y = X^H = (X^*)^T$ , где  $X^*$  - это комплексное сопряжение матрицы X, а  $X^T$  транспонированная матрица X.

Пример

```
X=eye(2)*(3+i)  В о з в р а щ а е т
```

<b>3+j1</b>	<b>0</b>
<b>0</b>	<b>3+j1</b>

Затем

```
Y=adjoint(X)  В о з в р а щ а е т
```

<b>3-j1</b>	<b>0</b>
<b>0</b>	<b>3-j1</b>

См. также

```
transpose(), conj()
```

# array()

Выводит единичные элементы.

## Синтаксис

Функция “array()” – это неявная команда. Таким образом, обычно используется соответствующее выражение (“preferred”).

Syntax	Preferred	Alternative	Preferred	Alternative
1	$y = VM[i,j]$	$y = \text{array}(VM,i,j)$		
2	$y = M[i,j]$	$y = \text{array}(M,i,j)$		
3	$y = VM[k]$	$y = \text{array}(VM,k)$		
4	$y = v[i]$	$y = \text{array}(v,i)$	$y = v[r]$	$y = \text{array}(v,r)$
5	$y = v[i,r]$	$y = \text{array}(v,i,r)$	$y = v[r,j]$	$y = \text{array}(v,r,j)$
	$y = v[i,j]$	$y = \text{array}(v,i,j)$	$y = v[r1,r2]$	$y = \text{array}(v,r1,r2)$
6	$y = s[i]$	$y = \text{array}(s,i)$		

## Аргументы

Name	Type	Def. Range	Required
VM	$\mathbb{R}^{m \times n \times p}, \mathbb{C}^{m \times n \times p}$	$]-\infty, +\infty[$	$\sqrt{(\text{Syntax 1 and 3})}$
M	$\mathbb{R}^{m \times n}, \mathbb{C}^{m \times n}$	$]-\infty, +\infty[$	$\sqrt{(\text{Syntax 2})}$
v	$\mathbb{R}^n, \mathbb{C}^n$	$]-\infty, +\infty[$	$\sqrt{(\text{Syntax 4 and 5})}$
r, r1, r2	Range $xs : xe$	$0 \leq xs \leq n-1, xs \leq xe \leq n-1$	$\sqrt{(\text{Syntax 4 and 5})}$
i	$\mathbb{N}$	$0 \leq i \leq m-1$	$\sqrt{(\text{Syntax 1, 2, 4, 5, 6})}$
j	$\mathbb{N}$	$0 \leq j \leq n-1$	$\sqrt{(\text{Syntax 1, 2, 5})}$
k	$\mathbb{N}$	$0 \leq k \leq p-1$	$\sqrt{(\text{Syntax 3})}$
s	String	Arbitrary characters	$\sqrt{(\text{Syntax 6})}$

## Описание

Эта функция выводит действительный или комплексный вектор матриц, матрицы и векторы или строки. Пожалуйста, обратитесь к следующей таблице возвращаемых значений:

Syntax	Argument 1	Argument 2	Argument 3	Result
y=VM[i,j]	$VM = (x_{ijk})$	$i \in \mathbb{N}$	$j \in \mathbb{N}$	Vector $(x_{ij1}, \dots, x_{ijK})$
y=M[i,j]	$M = (x_{ij})$	$i \in \mathbb{N}$	$j \in \mathbb{N}$	Number $x_{ij}$
y=VM[k]	$VM = (x_{ijk})$	$k \in \mathbb{N}$		Matrix $\begin{pmatrix} x_{11k} & \dots & x_{1nk} \\ \vdots & \ddots & \vdots \\ x_{m1k} & \dots & x_{mnk} \end{pmatrix}$
y=v[i]	$v = (v_i)$	$i \in \mathbb{N}$		Number $v_i$
y=v[xs:xe]	$v = (v_i)$	$xs, \dots, xe$		Vector $(v_{xs}, \dots, v_{xe})$
y=v[i,xs:xe]	$v = (v_i)$	$i \in \mathbb{N}$	$xs, \dots, xe$	Vector $(v_{xs}, \dots, v_{xe})$
y=v[xs:xe,j]	$v = (v_i)$	$xs, \dots, xe$	$xs, \dots, xe$	Vector $(v_{xs}, \dots, v_{xe})$
y=v[i,j]	$v = (v_i)$	$i \in \mathbb{N}$	$xs, \dots, xe$	Vector $(v_{xs}, \dots, v_{xe})$
y=v[xs1:xe1, xs2:xe2]	$v = (v_i)$	$xs1, \dots, xe1$	$xs2, \dots, xe2$	Vector $(v_{xs}, \dots, v_{xe})$
y=s[i]	$s = (s_i)$	$i \in \mathbb{N}$		Character $s_i$

Вновь, v указывает на вектор, M матрицу, VM вектор матриц, s вектор символов, а xs, xs1, xs2, xe, xe1, xe2 диапазон ограничителей.

### Пример

`v=linspace(1,2,4)` Возвращает 1, 1.33, 1.67, 2.

Затем

`y=v[3]` Возвращает 2.

**См. также**

**det()**

Определитель матрицы.

**Синтаксис**

y=det (X)

**Аргументы**

Name	Type	Def. Range	Required
X	$\mathbb{R}^{n \times n}, \mathbb{C}^{n \times n}, \mathbb{R}^{m \times n \times p}, \mathbb{C}^{m \times n \times p}$	$]-\infty, +\infty[$	$\surd$

**Описание**

Эта функция вычисляет определитель квадратной n x n матрицы X. Результат либо действительное, либо комплексное число.

**Пример**

X=eye(2)\*3    В о з в р а щ а е т

3	0
0	3

Затем

y=det(X)    В о з в р а щ а е т    9.

**См. также**

eye ( )

**inverse()**

**Инверсия матрицы.**

**Синтаксис**

```
Y=inverse(X)
```

**Аргументы**

Name	Type	Def. Range	Required
X	$\mathbb{R}^{n \times n}, \mathbb{C}^{n \times n}, \mathbb{R}^{m \times n \times p}, \mathbb{C}^{m \times n \times p}$	$] - \infty, + \infty[$	✓

**Описание**

Функция инвертирует квадратную n x n матрицу X. Сгенрированная инверсная матрица Y выполняет уравнение

$X \cdot Y = X \cdot X^{-1} = I$ , где “ $\cdot$ ” обозначает матричное умножение, а “I” единичная матрица.

Матрица X должна быть регулярной, что означает, что ее определитель  $\Delta \neq 0$ .

**Пример**

```
X=eye(2)*3  В о з в р а щ а е т
```

3	0
0	3

Затем

```
Y=inverse(X)  В о з в р а щ а е т
```

0.33 3	0
0	0.33 3

**См. также**

```
transpose(), eye(), det()
```

## transpose()

Транспозиция матрицы.

### Синтаксис

$Y = \text{transpose}(X)$

### Аргументы

Name	Type	Def. Range	Required
X	$\mathbb{R}^{m \times n}, \mathbb{C}^{m \times n}, \mathbb{R}^{m \times n \times p}, \mathbb{C}^{m \times n \times p}$	$]-\infty, +\infty[$	✓

### Описание

Эта функция транспонирует  $m \times n$  матрицу  $X$ , которая эквивалентна матрице с замененными строками и столбцами согласно

$$Y = X^T = (x_{ij})^T = (x_{ji}) \text{ с } 1 \leq i \leq m, 1 \leq j \leq n$$

Сгенерированная матрица  $Y$  – это матрица  $n \times m$ .

### Пример

$X = \text{eye}(2) * 3$  В о з в р а щ а е т

3	0
0	3

затем

$Y = \text{transpose}(X)$  В о з в р а щ а е т

3	0
0	3

См. также

`eye()`, `inverse()`

## 13.4.2 Элементарные математические функции

Базовые действительные и комплексные функции

### abs()

Абсолютное значение.

### Синтаксис

```
y=abs(x)
```

Аргументы

Name	Type	Def. Range	Required
x	$\mathbb{R}, \mathbb{C}, \mathbb{R}^n, \mathbb{C}^n, \mathbb{R}^{m \times n}, \mathbb{C}^{m \times n}, \mathbb{R}^{m \times n \times p}, \mathbb{C}^{m \times n \times p}$	$] -\infty, +\infty[$	$\sqrt{\phantom{x}}$

Описание

Эта функция вычисляет абсолютное значение действительного или комплексного числа, вектора или матрицы.

Для  $x \in \mathbb{R}$ :  $y = \begin{cases} x & \text{for } x \geq 0 \\ -x & \text{for } x < 0 \end{cases}$

Для  $\mathbb{C} \ni x := a + i b \wedge a, b \in \mathbb{R}$ :  $y = \sqrt{a^2 + b^2}$

Для x как вектора или матрицы два уравнения выше применяются к компонентам x.

Примеры

```
y=abs(-3)  В о з в р а щ а е т  3,  
y=abs(-3+4*i)  В о з в р а щ а е т  5.
```

См. также

```
mag(), norm(), real(), imag(), conj(), phase(), arg()
```

# angle()

Фазовый угол в радианах комплексного числа. Синоним “arg”.

## Синтаксис

```
y=angle(x)
```

## См. также

```
abs(), mag(), norm(), real(), imag(), conj(), phase(), arg()
```

# arg()

Фазовый угол в радианах комплексного числа.

## Синтаксис

```
y=arg(x)
```

## Аргументы

Name	Type	Def. Range	Required
x	$\mathbb{R}, \mathbb{C}, \mathbb{R}^n, \mathbb{C}^n, \mathbb{R}^{m \times n}, \mathbb{C}^{m \times n}, \mathbb{R}^{m \times n \times p}, \mathbb{C}^{m \times n \times p}$	$] - \infty, + \infty [$	✓

## Описание

Эта функция возвращает фазовый угол в градусах действительного или комплексного числа, вектора или матрицы.

Для  $x \in \mathbb{R}$ :  $y = \begin{cases} 0 & \text{for } x \geq 0 \\ \pi & \text{for } x < 0 \end{cases}$

Для  $\mathbb{C} \ni x := a + i b \wedge a, b \in \mathbb{R}$ :

Definition range	Result
$a > 0, b > 0$	$y = \arctan\left(\frac{b}{a}\right)$
$a < 0, b > 0$	$y = \arctan\left(\frac{b}{a}\right) + \pi$
$a < 0, b < 0$	$y = \arctan\left(\frac{b}{a}\right) - \pi$
$a > 0, b < 0$	$y = \arctan\left(\frac{b}{a}\right)$
$a = 0, b > 0$	$y = \frac{\pi}{2}$
$a > 0, b > 0$	$y = -\frac{\pi}{2}$
$a = 0, b = 0$	$y = 0$

В этом случае функция arctan() возвращает значение в радианах. Результат у фазовой функции находится в диапазоне  $[-\pi, +\pi]$ . Для x как вектора или матрицы два уравнения выше применяются к компонентам x.

## Примеры

```
y=arg(-3) Возвращает 3.14,  
y=arg(-3+4*i) Возвращает 2.21.
```

**См. также**

`abs()`, `mag()`, `norm()`, `real()`, `imag()`, `conj()`, `phase()`

conj()

Сопряжение комплексного числа.

Синтаксис

y=conj(x)

Аргументы

Name	Type	Def. Range	Required
x	$\mathbb{R}, \mathbb{C}, \mathbb{R}^n, \mathbb{C}^n, \mathbb{R}^{m \times n}, \mathbb{C}^{m \times n}, \mathbb{R}^{m \times n \times p}, \mathbb{C}^{m \times n \times p}$	$] - \infty, + \infty[$	<input checked="" type="checkbox"/>

Описание

Эта функция возвращает сопряжение действительного или комплексного числа, вектора или матрицы.

Для  $x \in \mathbb{R}$ :  $y = x$

Для  $\mathbb{C} \ni x := a + i b \wedge a, b \in \mathbb{R}$ :  $y = a - i b$

Для x как вектора или матрицы два уравнения выше применяются к компонентам x.

Пример

y=conj(-3+4\*i) В о з в р а щ а е т -3-4\*i.

См. также

abs(), mag(), norm(), real(), imag(), phase(), arg()

# deg2rad()

Конвертирует фазу из градусов в радианы.

## Синтаксис

```
y=deg2rad(x)
```

## Аргументы

Name	Type	Def. Range	Required
x	$\mathbb{R}, \mathbb{C}, \mathbb{R}^n, \mathbb{C}^n$	$] - \infty, + \infty [$	✓

## Описание

Эта функция конвертирует действительную фазу, комплексную фазу или фазу вектора, данные в градусах в радианы.

Для  $x \in \mathbb{R}$ :  $y = \frac{\pi}{180} x$

Для  $x \in \mathbb{C}$ :  $y = \frac{\pi}{180} Re \{x\}$

Для x как вектора два уравнения выше применяются к компонентам x.

## Пример

```
y=deg2rad(45) В о з в р а щ а е т 0.785.
```

## См. также

```
rad2deg(), phase(), arg()
```

imag()

Мнимое значение комплексного числа.

Синтаксис

```
y=imag(x)
```

Аргументы

Name	Type	Def. Range	Required
x	$\mathbb{R}, \mathbb{C}, \mathbb{R}^n, \mathbb{C}^n, \mathbb{R}^{m \times n}, \mathbb{C}^{m \times n}, \mathbb{R}^{m \times n \times p}, \mathbb{C}^{m \times n \times p}$	$] -\infty, +\infty[$	✓

Описание

Эта функция возвращает мнимое значение действительного или комплексного числа, вектора или матрицы.

Для  $x \in \mathbb{R}$ :  $y = 0$

Для  $\mathbb{C} \ni x := a + i b \wedge a, b \in \mathbb{R}$ :  $y = b$

Для x как вектора или матрицы два уравнения выше применяются к компонентам x.

Пример

```
y=imag(-3+4*i) В о з в р а щ а е т 4.
```

См. также

```
abs(), mag(), norm(), real(), conj(), phase(), arg()
```

mag()

Модуль комплексного числа.

Синтаксис

y=mag(x)

Аргументы

Name	Type	Def. Range	Required
x	$\mathbb{R}, \mathbb{C}, \mathbb{R}^n, \mathbb{C}^n, \mathbb{R}^{m \times n}, \mathbb{C}^{m \times n}, \mathbb{R}^{m \times n \times p}, \mathbb{C}^{m \times n \times p}$	$] - \infty, + \infty [$	<input checked="" type="checkbox"/>

Описание

Эта функция вычисляет модуль (абсолютное значение) действительного или комплексного числа, вектора или матрицы.

Для  $x \in \mathbb{R}$ :  $y = \begin{cases} x & \text{for } x \geq 0 \\ -x & \text{for } x < 0 \end{cases}$

Для  $\mathbb{C} \ni x := a + i b \wedge a, b \in \mathbb{R}$ :  $y = \sqrt{a^2 + b^2}$

Для x как вектора или матрицы два уравнения выше применяются к компонентам x.

Примеры

y=mag(-3) Возвращает 3,  
y=mag(-3+4\*i) Возвращает 5.

См. также

abs(), norm(), real(), imag(), conj(), phase(), arg()

**norm()**

**Квадрат абсолютного значения вектора.**

**Синтаксис**

```
y=norm(x)
```

**Аргументы**

Name	Type	Def. Range	Required
x	$\mathbb{R}, \mathbb{C}, \mathbb{R}^n, \mathbb{C}^n$	$] - \infty, + \infty [$	$\surd$

**Описание**

Эта функция возвращает квадрат абсолютного значения действительного или комплексного числа, вектора или матрицы.

Для  $x \in \mathbb{R}$ :  $y = x^2$

Для  $\mathbb{C} \ni x := a + i b \wedge a, b \in \mathbb{R}$ :  $y = a^2 + b^2$

Для x как вектора или матрицы два уравнения выше применяются к компонентам x.

**Пример**

```
y=norm(-3+4*i)  В о з в р а щ а е т  25.
```

**См. также**

```
abs(), mag(), real(), imag(), conj(), phase(), arg()
```

phase()

Фазовый угол комплексного числа в градусах.

Синтаксис

```
y=phase(x)
```

Аргументы

Name	Type	Def. Range	Required
x	$\mathbb{R}, \mathbb{C}, \mathbb{R}^n, \mathbb{C}^n, \mathbb{R}^{m \times n}, \mathbb{C}^{m \times n}, \mathbb{R}^{m \times n \times p}, \mathbb{C}^{m \times n \times p}$	$]-\infty, +\infty[$	$\surd$

Описание

Эта функция возвращает фазовый угол в градусах действительного или комплексного числа, вектора или матрицы.

Для  $x \in \mathbb{R}$ :  $y = \begin{cases} 0 & \text{for } x \geq 0 \\ 180 & \text{for } x < 0 \end{cases}$

Для  $\mathbb{C} \ni x := a + i b \wedge a, b \in \mathbb{R}$ :

Definition range	Result
$a > 0, b > 0$	$y = \arctan\left(\frac{b}{a}\right)$
$a < 0, b > 0$	$y = \arctan\left(\frac{b}{a}\right) + 180$
$a < 0, b < 0$	$y = \arctan\left(\frac{b}{a}\right) - 180$
$a > 0, b < 0$	$y = \arctan\left(\frac{b}{a}\right)$
$a = 0, b > 0$	$y = 90$
$a > 0, b = 0$	$y = -90$
$a = 0, b = 0$	$y = 0$

В этом случае функция arctan() возвращает значения в градусах. Результат функции phase находится в диапазоне  $[-180, +180]$ . Для x как вектора или матрицы два уравнения выше применяются к компонентам x.

Примеры

```
y=phase(-3) возвращает 180,  
y=phase(-3+4*i) возвращает 127.
```

См. также

```
abs(), mag(), norm(), real(), imag(), conj(), arg()
```

## polar()

Преобразовывает из полярных координат в комплексное число.

### Синтаксис

`c=polar(a,p)`

### Аргументы

Name	Type	Def. Range	Required
a	$\mathbb{R}^n, \mathbb{C}^n$	$] - \infty, + \infty [$	✓
p	$\mathbb{R}^n, \mathbb{C}^n$	$] - \infty, + \infty [$	✓

### Описание

Эта функция преобразует точку, заданную в полярных координатах (амплитуда и фаза p в градусах) в комплексный вид соответствующего комплексного числа:

$$x + iy = ae^{ip} = a \cos p + ia \sin p$$

Для a или p как векторов уравнение выше применяется к компонентам a или p.

### Пример

`c=polar(3,45)` В о з в р а щ а е т 2.12+j2.12.

### См. также

`abs()`, `mag()`, `norm()`, `real()`, `imag()`, `conj()`, `phase()`, `arg()`, `exp()`, `cos()`, `sin()`

rad2deg()

Конвертирует фазу из градусов в радианы.

Синтаксис

y=rad2deg(x)

Аргументы

Name	Type	Def. Range	Required
x	$\mathbb{R}, \mathbb{C}, \mathbb{R}^n, \mathbb{C}^n$	$] - \infty, + \infty [$	✓

Описание

Эта функция преобразует действительную фазу, комплексную фазу или векторную фазу, данные в радианах, в градусы.

Для  $x \in \mathbb{R}$ :  $y = \frac{180}{\pi} x$

Для  $x \in \mathbb{C}$ :  $y = \frac{180}{\pi} Re \{x\}$

Для x как вектора два уравнения выше применяются к компонентам x.

Пример

y=deg2rad(0.785) В о з в р а щ а е т 45.

См. также

deg2rad(), phase(), arg()

**real()**

Действительное значение комплексного числа.

**Синтаксис**

```
y=real(x)
```

**Аргументы**

Name	Type	Def. Range	Required
x	$\mathbb{R}, \mathbb{C}, \mathbb{R}^n, \mathbb{C}^n, \mathbb{R}^{m \times n}, \mathbb{C}^{m \times n}, \mathbb{R}^{m \times n \times p}, \mathbb{C}^{m \times n \times p}$	$] -\infty, +\infty[$	$\checkmark$

**Описание**

Эта функция возвращает действительное значение действительного или комплексного числа, вектора или матрицы.

Для  $x \in \mathbb{R}$ :  $y = x$

Для  $\mathbb{C} \ni x := a + i b \wedge a, b \in \mathbb{R}$ :  $y = a$

Для x как вектор или матрицы два уравнения выше применяются к компонентам x.

**Пример**

```
y=real(-3+4*i) В о з в р а щ а е т -3.
```

**См. также**

```
abs(), mag(), norm(), imag(), conj(), phase(), arg()
```

## signum()

Сигнум-функция.

### Синтаксис

`y=signum(x)`

### Аргументы

Name	Type	Def. Range	Required
x	$\mathbb{R}, \mathbb{C}, \mathbb{R}^n, \mathbb{C}^n$	$]-\infty, +\infty[$	✓

### Описание

Эта функция вычисляет действительного или комплексного числа или вектора.

Для  $x \in \mathbb{R}$ :  $y = \begin{cases} 1 & \text{for } x > 0 \\ 0 & \text{for } x = 0 \\ -1 & \text{for } x < 0 \end{cases}$

Для  $x \in \mathbb{C}$ :  $y = \begin{cases} \frac{x}{|x|} & \text{for } x \neq 0 \\ 0 & \text{for } x = 0 \end{cases}$

Для x как вектора два уравнения выше применяются к компонентам x.

### Примеры

`y=signum(-4)` В о з в р а щ а е т `-1`,  
`y=signum(3+4*i)` В о з в р а щ а е т `0.6+j0.8`.

### См. также

`abs()`, `sign()`

**sign()**

**Знаковая функция.**

**Синтаксис**

```
y=sign(x)
```

**Аргументы**

Name	Type	Def. Range	Required
x	$\mathbb{R}, \mathbb{C}, \mathbb{R}^n, \mathbb{C}^n$	$]-\infty, +\infty[$	✓

**Описание**

Эта функция вычисляет знак действительного или комплексного числа или вектора.

Для  $x \in \mathbb{R}$ :  $y = \begin{cases} 1 & \text{for } x \geq 0 \\ -1 & \text{for } x < 0 \end{cases}$

Для  $x \in \mathbb{C}$ :  $y = \begin{cases} \frac{x}{|x|} & \text{for } x \neq 0 \\ 1 & \text{for } x = 0 \end{cases}$

Для x как вектор два уравнения выше применяются к компонентам x.

**Примеры**

```
y=sign(-4) Возвращает -1,  
y=sign(3+4*i) Возвращает 0.6+j0.8.
```

**См. также**

```
abs(), signum()
```

**sqr()**

**Квадрат числа.**

**Синтаксис**

$y=\text{sqr}(x)$

**Аргументы**

Name	Type	Def. Range	Required
x	$\mathbb{R}, \mathbb{C}, \mathbb{R}^n, \mathbb{C}^n$	$]-\infty, +\infty[$	$\sqrt{\phantom{x}}$

**Описание**

Эта функция вычисляет квадрат действительного или мнимого числа или вектора.

$y = x^2$

Для x как вектора уравнение выше применяется к компонентам x.

**Примеры**

$y=\text{sqr}(-4)$  В о з в р а щ а е т 16,  
 $y=\text{sqr}(3+4*i)$  В о з в р а щ а е т -7+j24.

**См. также**

`sqrt()`

sqrt()

Квадратный корень.

Синтаксис

y= sqrt (x)

Аргументы

Name	Type	Def. Range	Required
x	$\mathbf{R, C, R^n, C^n}$	$] - \infty, + \infty[$	$\sqrt{\phantom{x}}$

Описание

Эта функция вычисляет квадратный корень действительного или комплексного числа или вектора.

Для  $x \in \mathbb{R}$ :  $y = \begin{cases} \sqrt{x} & for\ x \geq 0 \\ i\sqrt{-x} & for\ x < 0 \end{cases}$

Для  $x \in \mathbb{C}$ :  $y = \sqrt{|x|} e^{i\frac{\varphi}{2}}$  with  $\varphi = \arg(x)$

Для x как вектора два уравнения выше применяются к компонентам x.

Примеры

y=sqrt(-4) В о з в р а щ а е т 0+j2,  
y=sqrt(3+4\*i) В о з в р а щ а е т 2+j1.

См. также

sqr()

## unwrap()

Разворачивает фазу вектора в радианах.

### Синтаксис

```
y=unwrap(x)
y=unwrap(x, t)
```

### Аргументы

Name	Type	Def. Range	Required	Default
x	$\mathbb{R}^n, \mathbb{C}^n$	$]-\infty, +\infty[$	✓	
t	$\mathbb{R}$	$]-\infty, +\infty[$		$\pi$

### Описание

Эта функция разворачивает фазу вектора x во избежание фазового скачка, если два последовательных значения x, отличающихся более, чем допустимым t,  $\pm 2\pi$  (в зависимости от знака разницы) добавляются к текущему элементу x. Предопределенное значение дополнительного параметра t –  $\pi$ .

### Примеры

```
y=unwrap(3.15*linspace(-2,2,5)) Возвращает -6.3, -9.43, -12.6, -15.7, -18.8,
y=unwrap(2*linspace(-2,2,5),1) Возвращает -4, -8.28, -12.6, -16.8, -21.1,
y=unwrap(2*linspace(-2,2,5),3) Возвращает -4, -2, 0, 2, 4.
```

### См. также

abs(), mag(), norm(), real(), imag(), conj(), phase(), arg()

## Экспоненциальная и логарифмические функции

### exp()

Экспоненциальная функция.

#### Синтаксис

`y=exp(x)`

#### Аргументы

Name	Type	Def. Range	Required
x	$\mathbb{R}, \mathbb{C}, \mathbb{R}^n, \mathbb{C}^n$	$]-\infty, +\infty[$	✓

#### Описание

Эта функция вычисляет экспоненциальную функцию действительного или комплексного числа или вектора.

Для  $x \in \mathbb{R}$ :  $y = e^x$

Для  $\mathbb{C} \ni x := a + i b \wedge a, b \in \mathbb{R}$ :  $y = e^x = e^{a+ib} = e^a (\cos b + i \sin b)$

Для x как вектор два уравнения выше применяются к компонентам x.

#### Примеры

`y=exp(-4)` Возвращает 0.0183,

`y=exp(3+4*i)` Возвращает -13.1-j15.2.

#### См. также

`ln()`, `log10()`, `log2()`, `cos()`, `sin()`

**log10()**

Десятичный логарифм.

**Синтаксис**

y=log10(x)

**Аргументы**

Name	Type	Def. Range	Required
x	$\mathbb{R}, \mathbb{C}, \mathbb{R}^n, \mathbb{C}^n$	$]-\infty, +\infty[ \setminus \{0\}$	✓

**Описание**

Эта функция вычисляет основное значение десятичного логарифма (по основанию 10) действительного или комплексного числа или вектора.

Для  $x \in \mathbb{R}$ :  $y = \begin{cases} \frac{\ln(x)}{\ln(10)} & for\ x > 0 \\ \frac{\ln(-x)}{\ln(10)} + i \frac{\pi}{\ln(10)} & for\ x < 0 \end{cases}$

Для  $x \in \mathbb{C}$ :  $y = \frac{\ln(|x|)}{\ln(10)} + i \frac{\arg(x)}{\ln(10)}$

Для x как вектор два уравнения выше применяются к компонентам x.

**Примеры**

y=log10(-4) Возвращает 0.602+j1.36,  
y=log10(3+4\*i) Возвращает 0.699+j0.403.

**См. также**

ln(), log2(), exp(), arg()

log2()

Двоичный логарифм.

Синтаксис

```
y=log2(x)
```

Аргументы

Name	Type	Def. Range	Required
x	$\mathbb{R}, \mathbb{C}, \mathbb{R}^n, \mathbb{C}^n$	$]-\infty, +\infty[\setminus \{0\}$	✓

Описание

Эта функция вычисляет основное значение двоичного логарифма (основание 2) действительного или комплексного числа или вектора.

Для  $x \in \mathbb{R}$ : 
$$y = \begin{cases} \frac{\ln(x)}{\ln(2)} & for\ x > 0 \\ \frac{\ln(-x)}{\ln(2)} + i \frac{\pi}{\ln(2)} & for\ x < 0 \end{cases}$$

Для  $x \in \mathbb{C}$ : 
$$y = \frac{\ln(|x|)}{\ln(2)} + i \frac{\arg(x)}{\ln(2)}$$

Для x как вектора два уравнения выше применяются к компонентам x.

Примеры

```
y=log2(-4)  В о з в р а щ а е т  2+j4.53,  
y=log2(3+4*i)  В о з в р а щ а е т  2.32+j1.34.
```

См. также

```
ln(), log10(), exp(), arg()
```

**ln()**

**Натуральный логарифм (основание e).**

**Синтаксис**

```
y=ln(x)
```

**Аргументы**

Name	Type	Def. Range	Required
x	$\mathbb{R}, \mathbb{C}, \mathbb{R}^n, \mathbb{C}^n$	$]-\infty, +\infty[ \setminus \{0\}$	$\surd$

**Описание**

Эта функция вычисляет основное значение натурального логарифма (основание e) действительного или комплексного числа или вектора.

Для  $x \in \mathbb{R}$ :  $y = \begin{cases} \ln(x) & \text{for } x > 0 \\ \ln(-x) & \text{for } x < 0 \end{cases}$

Для  $x \in \mathbb{C}$ :  $y = \ln(|x|) + i \arg(x)$

Для x как вектор два уравнения выше применяются к компонентам x.

**Примеры**

```
y=ln(-4)  В о в р а щ а е т  1.39+j3.14,  
y=ln(3+4*i)  В о з в р а щ а е т  1.61+j0.927.
```

**См. также**

```
log2(), log10(), exp(), arg()
```

# Тригонометрия

cos()

Функция косинуса.

Синтаксис

y=cos(x)

Аргументы

Name	Type	Def. Range	Required
x	$\mathbb{R}, \mathbb{C}, \mathbb{R}^n, \mathbb{C}^n$	$] -\infty, +\infty[$	✓

Описание

Эта функция вычисляет косинус действительного или комплексного числа или вектора.

Для  $x \in \mathbb{R}$ :  $y = \cos(x)$  with  $y \in [-1, 1]$

Для  $x \in \mathbb{C}$ :  $y = \frac{1}{2} (\exp(i x) + \exp(-i x))$

Для x как вектор два уравнения выше применяются к компонентам x.

Примеры

y=cos(-0.5) Возвращает 0.878,  
y=cos(3+4\*i) Возвращает -27.0-j3.85.

См. также

sin(), tan(), arccos()

**cosec()**

**Косеканс.**

**Синтаксис**

```
y=cosec(x)
```

**Аргументы**

Name	Type	Def. Range	Required
x	$\mathbb{R}, \mathbb{C}, \mathbb{R}^n, \mathbb{C}^n$	$]-\infty, +\infty[ \setminus \{k\pi\}, k \in \mathbb{Z}$	$\checkmark$

**Описание**

Эта функция вычисляет косеканс действительного или комплексного числа или вектора.

$$y = \operatorname{cosec} x = \frac{1}{\sin x}$$

Для x как вектор уравнение выше применяется к компонентам x.

**Пример**

```
y=cosec(1)  В о з в р а щ а е т  1.19.
```

**См. также**

```
sin(), sec()
```

**cot()**

**Функция котангенса.**

**Синтаксис**

```
y=cot (x)
```

**Аргументы**

Name	Type	Def. Range	Required
x	$\mathbb{R}, \mathbb{C}, \mathbb{R}^n, \mathbb{C}^n$	$] -\infty, +\infty[ \setminus \{k\pi\}, k \in \mathbb{Z}$	$\checkmark$

**Описание**

Эта функция вычисляет котангенс действительного или комплексного числа или вектора.

Для  $x \in \mathbb{R}$ :  $y = \frac{1}{\tan (x)}$  with  $y \in [-\infty, +\infty]$

Для  $x \in \mathbb{C}$ :  $y = i \left( \frac{\exp (i x)^2 + 1}{\exp (i x)^2 - 1} \right)$

Для x как вектор два уравнения выше применяются к компонентам x.

**Примеры**

```
y=cot(-0.5) В о з в р а щ а е т -1.83,  
y=cot(3+4*i) В о з в р а щ а е т -0.000188-j1.
```

**См. также**

```
tan(), sin(), cos(), arctan(), arccot()
```

**sec()**

**Секанс.**

**Синтаксис**

$y=\sec (x)$

**Аргументы**

Name	Type	Def. Range	Required
x	$\mathbb{R}, \mathbb{C}, \mathbb{R}^n, \mathbb{C}^n$	$]-\infty, +\infty[\setminus \{(k+\frac{1}{2})\pi\}, k\in \mathbb{Z}$	$\surd$

**Описание**

Эта функция вычисляет секанс действительного или комплексного числа или вектора.

$$y = \sec x = \frac{1}{\cos x}$$

Для x как вектор уравнение выше применяется к компонентам x.

**Пример**

$y=\sec(0)$  В о з в р а щ а е т 1.

**См. также**

$\cos()$ ,  $\operatorname{cosec}()$

**sin()**

**Функция синуса.**

**Синтаксис**

y=sin(x)

**Аргументы**

Name	Type	Def. Range	Required
x	$\mathbb{R}, \mathbb{C}, \mathbb{R}^n, \mathbb{C}^n$	$] - \infty, + \infty [$	✓

**Описание**

Эта функция вычисляет синус действительного или комплексного числа или вектора.

Для  $x \in \mathbb{R}$ :  $y = \sin(x)$  with  $y \in [-1, 1]$

Для  $x \in \mathbb{C}$ :  $y = \frac{1}{2} i \left( \exp(-i x) - \exp(i x) \right)$

Для x как вектора два уравнения выше применяются к компонентам x.

**Примеры**

y=sin(-0.5) В о з в р а щ а е т -0.479,  
y=sin(3+4\*i) В о з в р а щ а е т 3.85-j27.

**См. также**

cos(), tan(), arcsin()

**tan()**

**Функция тангенса.**

**Синтаксис**

```
y=tan(x)
```

**Аргументы**

Name	Type	Def. Range	Required
x	$\mathbb{R}, \mathbb{C}, \mathbb{R}^n, \mathbb{C}^n$	$]-\infty, +\infty[\setminus \{(k+\frac{1}{2})\pi\}, k \in \mathbb{Z}$	$\surd$

**Описание**

Эта функция вычисляет тангенс действительного или комплексного числа или вектора.

Для  $x \in \mathbb{R}$ :  $y = \tan(x)$  with  $y \in [-\infty, +\infty]$

Для  $x \in \mathbb{C}$ :  $y = -i \left( \frac{\exp(i x)^2 - 1}{\exp(i x)^2 + 1} \right)$

Для x как вектора два уравнения выше применяются к компонентам x.

**Примеры**

```
y=tan(-0.5) В о з в р а щ а е т -0.546,  
y=tan(3+4*i) В о з в р а щ а е т -0.000187+j0.999.
```

**См. также**

```
cot(), sin(), cos(), arctan(), arccot()
```

## Обратные тригонометрические функции

### arccos()

Арккосинус (известный также как “обратный косинус”).

#### Синтаксис

`y=arccos(x)`

#### Аргументы

Name	Type	Def. Range	Required
x	$\mathbb{R}, \mathbb{C}, \mathbb{R}^n, \mathbb{C}^n$	$[-1, +1]$	✓

#### Описание

Эта функция вычисляет основное значение арккосинуса действительного или комплексного числа или вектора.

Для  $x \in \mathbb{R}$ :  $y = \arccos(x)$  with  $y \in [0, \pi]$

Для  $x \in \mathbb{C}$ :  $y = -i \ln(x + \sqrt{x^2 - 1})$

Для x как вектора два уравнения выше применяются к компонентам x.

#### Примеры

`y=arccos(-1)` В о з в р а щ а е т 3.14,  
`y=arccos(3+4*i)` В о з в р а щ а е т 0.937-j2.31.

#### См. также

`cos()`, `arcsin()`, `arctan()`, `arccot()`

**arccot()**

**Арккотангенс.**

**Синтаксис**

```
y=arccot(x)
```

**Аргументы**

Name	Type	Def. Range	Required
x	$\mathbb{R}, \mathbb{C}, \mathbb{R}^n, \mathbb{C}^n$	$] - \infty, + \infty [$	✓

**Описание**

Эта функция вычисляет основное значение арккотангенса действительного или комплексного числа или вектора.

Для  $x \in \mathbb{R}$ :  $y = \text{arccot}(x)$  with  $y \in [0, \pi]$

Для  $x \in \mathbb{C}$ :  $y = \frac{i}{2} \ln \left( \frac{x - i}{x + i} \right)$

Для x как вектора два уравнения выше применяются к компонентам x.

**Примеры**

```
y=arccot(-1) В о з в р а щ а е т  2.36,  
y=arccot(3+4*i) В о з в р а щ а е т  0.122-j0.159.
```

**См. также**

```
cot(), tan(), arccos(), arcsin(), arctan()
```

**arcsin()**

Арсинус (также известный как “обратный синус”).

**Синтаксис**

```
y=arcsin(x)
```

**Аргументы**

Name	Type	Def. Range	Required
x	$\mathbb{R}, \mathbb{C}, \mathbb{R}^n, \mathbb{C}^n$	$[-1, +1]$	$\checkmark$

**Описание**

Эта функция вычисляет основное значение арксинуса действительного или комплексного числа или вектора.

Для  $x \in \mathbb{R}$ :  $y = \arcsin(x)$  with  $y \in \left[-\frac{\pi}{2}, \frac{\pi}{2}\right]$

Для  $x \in \mathbb{C}$ :  $y = -i \ln \left[ i x + \sqrt{1 - x^2} \right]$

Для x как вектора два уравнения выше применяются к компонентам x.

**Примеры**

```
y=arcsin(-1) В о з в р а щ а е т -1.57,  
y=arcsin(3+4*i) В о з в р а щ а е т 0.634+j2.31.
```

**См. также**

sin(), arccos(), arctan(), arccot()

**arctan()**

Арктангенс (также известный как “обратный тангенс”).

**Синтаксис**

```
z=arctan(x)
z=arctan(y,x)
```

**Аргументы**

Name	Type	Def. Range	Required
x	$\mathbb{R}, \mathbb{C}, \mathbb{R}^n, \mathbb{C}^n$	$]-\infty, +\infty[$	✓
y	$\mathbb{R}, \mathbb{C}, \mathbb{R}^n, \mathbb{C}^n$	$]-\infty, +\infty[$	

**Описание**

Для первого синтаксиса (  $z=\arctan(x)$  ), эта функция вычисляет основное значение арктангенса действительного или комплексного числа или вектора.

Для  $x \in \mathbb{R}$ :  $y = \arctan(x)$  with  $y \in [-\frac{\pi}{2}, \frac{\pi}{2}]$

Для  $x \in \mathbb{C}$ :  $y = -\frac{1}{2}i \ln \left[ \frac{2i}{x+i} - 1 \right]$

Для x как вектор два уравнения выше применяются к компонентам x.

Если второй синтаксис (  $z=\arctan(y, x)$  ) встречается в приложении, выражение

$z = \pm \arctan(y/x)$

(с функцией arctan(), определенной выше) вычисляется. Знак z определяется из

$$\text{sign}(z)=\left\{ \begin{array}{ll} + & \text{for } Re\{x\} > 0 \\ - & \text{for } Re\{x\} < 0 \end{array} \right.$$

Заметьте, что второй синтаксис в случае  $x = y = 0$  не определен.

## Примеры

`z=arctan(-1)` Возвращает `-0.785`,  
`z=arctan(3+4*i)` Возвращает `1.45+j0.159`,  
`z=arctan(1,1)` Возвращает `0.785`.

## См. также

`tan()`, `arccos()`, `arcsin()`, `arccot()`

**Гиперболические функции**

**cosh()**

**Гиперболический косинус.**

**Синтаксис**

y=cosh(x)

**Аргументы**

Name	Type	Def. Range	Required
x	$\mathbb{R}, \mathbb{C}, \mathbb{R}^n, \mathbb{C}^n$	$] - \infty, + \infty [$	✓

**Описание**

Эта функция вычисляет гиперболический косинус действительного или комплексного числа или вектора.

$$y = \frac{1}{2} (e^x + e^{-x})$$

Для x как вектор уравнение выше применяется к компонентам x.

**Примеры**

y=cosh(-1) В о з в р а щ а е т 1.54,  
y=cosh(3+4\*i) В о з в р а щ а е т -6.58-j7.58.

**См. также**

exp(), sinh(), tanh(), cos()

**cosech()**

**Гиперболический косеканс.**

**Синтаксис**

```
y=cosech(x)
```

**Аргументы**

Name	Type	Def. Range	Required
x	$\mathbb{R}, \mathbb{C}, \mathbb{R}^n, \mathbb{C}^n$	$]-\infty, +\infty[\setminus \{0\}$	✓

**Описание**

Эта функция вычисляет гиперболический косеканс действительного или комплексного числа или вектора.

$$y = \frac{1}{\sinh x}$$

Для x как вектор уравнение выше применяется к компонентам x.

**Примеры**

```
y=cosech(-1)  В о з в р а щ а е т  -0.851,  
y=cosech(3+4*i)  В о з в р а щ а е т  -0.0649+j0.0755.
```

**См. также**

```
exp(), sinh(), sech(), cosec()
```

**coth()**

**Гиперболический котангенс.**

**Синтаксис**

```
y=coth(x)
```

**Аргументы**

Name	Type	Def. Range	Required
x	$\mathbb{R}, \mathbb{C}, \mathbb{R}^n, \mathbb{C}^n$	$]-\infty, +\infty[\setminus \{0\}$	✓

**Описание**

Эта функция вычисляет гиперболический котангенс действительного или комплексного числа или вектора.

$$y = \frac{1}{\tanh x} = \frac{e^x + e^{-x}}{e^x - e^{-x}}$$

Для x как вектор уравнение выше применяется к компонентам x.

**Примеры**

```
y=coth(-1) Возвращает -1.31,  
y=coth(3+4*i) Возвращает 0.999-j0.0049.
```

**См. также**

```
exp(), cosh(), sinh(), tanh(), tan()
```

**sech()**

**Гиперболический секанс.**

**Синтаксис**

```
y=sech(x)
```

**Аргументы**

Name	Type	Def. Range	Required
x	$\mathbb{R}, \mathbb{C}, \mathbb{R}^n, \mathbb{C}^n$	$]-\infty, +\infty[$	✓

**Описание**

Эта функция вычисляет гиперболический секанс действительного или комплексного числа или вектора.

$$y = \frac{1}{\cosh x}$$

Для x как вектор уравнение выше применяется к компонентам x.

**Примеры**

```
y=sech(-1) В о з в р а щ а е т  0.648,  
y=sech(3+4*i) В о з в р а щ а е т  -0.0653+j0.0752.
```

**См. также**

```
exp(), cosh(), cosech(), sec()
```

**sinh()**

**Гиперболический синус.**

**Синтаксис**

y=sinh(x)

**Аргументы**

Name	Type	Def. Range	Required
x	$\mathbb{R}, \mathbb{C}, \mathbb{R}^n, \mathbb{C}^n$	$] - \infty, + \infty [$	✓

**Описание**

Эта функция вычисляет гиперболический синус действительного или комплексного числа или вектора.

$$y = \frac{1}{2} (e^x - e^{-x})$$

Для x как вектор уравнение выше применяется к компонентам x.

**Примеры**

y=sinh(-1) В о з в р а щ а е т -1.18,  
y=sinh(3+4\*i) В о з в р а щ а е т -6.55-j7.62.

**См. также**

exp(), cosh(), tanh(), sin()

**tanh()**

**Гиперболический тангенс.**

**Синтаксис**

y=tanh(x)

Аргументы

Name	Type	Def. Range	Required
x	$\mathbb{R}, \mathbb{C}, \mathbb{R}^n, \mathbb{C}^n$	$] - \infty, + \infty [$	✓

**Описание**

Эта функция вычисляет гиперболический тангенс действительного или комплексного числа или вектора.

$$y = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$

Для x как вектор уравнение выше применяется к компонентам x.

**Примеры**

y=tanh(-1) В о з в р а щ а е т -0.762,  
y=tanh(3+4\*i) В о з в р а щ а е т 1+j0.00491.

**См. также**

exp(), cosh(), sinh(), coth(), tan()

## Обратные гиперболические функции

### arcosh()

Гиперболический арка-косинус.

#### Синтаксис

`y=arcosh(x)`

#### Аргументы

Name	Type	Def. Range	Required
x	$\mathbf{R}, \mathbf{C}, \mathbf{R}^n, \mathbf{C}^n$	$[1, +\infty[$	✓

#### Описание

Эта функция вычисляет гиперболический арка-косинус действительного или комплексного числа или вектора, функция обратна функции “cosh”.

$$y = \operatorname{arcosh} x = \ln \left( x + \sqrt{x^2 - 1} \right)$$

Для x как вектор уравнение выше применяется к компонентам x.

#### Примеры

`y=arcosh(1)` В о з в р а щ а е т 0,  
`y=arcosh(3+4*i)` В о з в р а щ а е т 2.31+j0.937.

#### См. также

`arsinh()`, `artanh()`, `cosh()`, `arccos()`, `ln()`, `sqrt()`

**arcoth()**

**Гиперболический ареа-котангенс.**

**Синтаксис**

```
y=arcoth(x)
```

**Аргументы**

Name	Type	Def. Range	Required
x	$\mathbb{R}, \mathbb{C}, \mathbb{R}^n, \mathbb{C}^n$	$]-\infty, -1[ \cup ]1, +\infty[$	✓

**Описание**

Эта функция вычисляет гиперболический ареа-котангенс действительного или комплексного числа или вектора, функция обратна “cotanh”.

$$y = \operatorname{arcoth} x = \frac{1}{2} \ln \left( \frac{x + 1}{x - 1} \right)$$

Для x как вектор уравнение выше применяется к компонентам x.

**Примеры**

```
y=arcoth(2) В о з в р а щ а е т  0.549,  
y=arcoth(3+4*i) В о з в р а щ а е т  0.118-j0.161.
```

**См. также**

```
arsinh(), arcosh(), tanh(), arctan(), ln(), sqrt()
```

arsinh()

**Гиперболический аресинус.**

**Синтаксис**

y=arsinh(x)

**Аргументы**

Name	Type	Def. Range	Required
x	$\mathbf{R}, \mathbf{C}, \mathbf{R}^n, \mathbf{C}^n$	$] - \infty, + \infty [$	✓

**Описание**

Эта функция вычисляет гиперболический аресинус действительного или комплексного числа или вектора, функция обратна “sinh”.

$$y = \operatorname{arsinh} x = \ln \left( x + \sqrt{x^2 + 1} \right)$$

Для x как вектор уравнение выше применяется к компонентам x.

**Примеры**

y=arsinh(1) В о з в р а щ а е т 0.881,  
y=arsinh(3+4\*i) В о з в р а щ а е т 2.3+j0.918.

**См. также**

arcosh(), artanh(), sinh(), arcsin(), ln(), sqrt()

**artanh()**

**Гиперболический арка-тангенс.**

**Синтаксис**

```
y=artanh(x)
```

**Аргументы**

Name	Type	Def. Range	Required
x	$\mathbb{R}, \mathbb{C}, \mathbb{R}^n, \mathbb{C}^n$	$] -1, +1[$	✓

**Описание**

Эта функция вычисляет гиперболический арка-тангенс действительного или комплексного числа или вектора, функция обратна “tanh”.

$$y = \operatorname{artanh} x = \frac{1}{2} \ln \left( \frac{1+x}{1-x} \right)$$

Для x как вектор уравнение выше применяется к компонентам x.

**Примеры**

```
y=artanh(0)  В о з в р а щ а е т  0,  
y=artanh(3+4*i)  В о з в р а щ а е т  0.118+ji1.41.
```

**См. также**

```
arsinh(), arcosh(), tanh(), arctan(), ln(), sqrt()
```

## Округление

### ceil()

Округлене до следующего большего целого.

#### Синтаксис

`y=ceil(x)`

#### Аргументы

Name	Type	Def. Range	Required
x	$\mathbb{R}, \mathbb{C}, \mathbb{R}^n, \mathbb{C}^n$	$]-\infty, +\infty[$	✓

#### Описание

Функция округляет действительное число  $x$  до следующего большего целого значения.

Если  $x$  – комплексное число, обе, действительная и мнимая части округляются. Для  $x$  как вектор операция выше применяется к компонентам  $x$ .

#### Примеры

`y=ceil(-3.5)` В о з в р а ш а е т `-3`,  
`y=ceil(3.2+4.7*i)` В о з в р а ш а е т `4+j5`.

#### См. также

`floor()`, `fix()`, `round()`

## **fix()**

**Округляет целую часть действительного числа.**

### **Синтаксис**

`y=fix(x)`

### **Аргументы**

Name	Type	Def. Range	Required
x	$\mathbb{R}, \mathbb{C}, \mathbb{R}^n, \mathbb{C}^n$	$]-\infty, +\infty[$	✓

### **Описание**

Функция отсекает десятичную часть действительного числа `x` и возвращает целое.

Если `x` – комплексное число, обе действительная и мнимая части округляются. Для `x` как вектор операция выше применяется к компонентам `x`.

### **Примеры**

`y=fix(-3.5)` В о з в р а щ а е т `-3`,  
`y=fix(3.2+4.7*i)` В о з в р а щ а е т `3+j4`.

### **См. также**

`ceil()`, `floor()`, `round()`

# floor()

Округляет до следующего меньшего целого.

## Синтаксис

```
y=floor(x)
```

## Аргументы

Name	Type	Def. Range	Required
x	$\mathbb{R}, \mathbb{C}, \mathbb{R}^n, \mathbb{C}^n$	$] -\infty, +\infty[$	✓

## Описание

Эта функция округляет действительное число x до следующего меньшего целого значения.

Если x – комплексное число, обе действительная и мнимая части округляются. Для x как вектор операция выше применяется к компонентам x.

## Примеры

```
y=floor(-3.5) Возвращает -4,  
y=floor(3.2+4.7*i) Возвращает 3+j4.
```

## См. также

```
ceil(), fix(), round()
```

## round()

Округление до ближайшего целого.

### Синтаксис

`y=round(x)`

### Аргументы

Name	Type	Def. Range	Required
x	$\mathbb{R}, \mathbb{C}, \mathbb{R}^n, \mathbb{C}^n$	$]-\infty, +\infty[$	✓

### Описание

Эта функция округляет действительное число  $x$  до ближайшего к нему целого значения.

Если  $x$  – комплексное число, обе действительная и мнимая части округляются. Для  $x$  как вектор операция выше применяется к компонентам  $x$ .

### Примеры

`y=round(-3.5)` Возвращает `-4`,  
`y=round(3.2+4.7*i)` Возвращает `3+j5`.

### См. также

`ceil()`, `floor()`, `fix()`

## Специальные математические функции

### besseli0()

Модифицированная функция Бесселя нулевого порядка.

#### Синтаксис

`i0=besseli0(x)`

#### Аргументы

Name	Type	Def. Range	Required
x	$\mathbb{R}, \mathbb{C}, \mathbb{R}^n, \mathbb{C}^n$	$] -\infty, +\infty[$	✓

#### Описание

Эта функция вычисляет модифицированную функцию Бесселя нулевого порядка действительного или комплексного числа или вектора.

$$i_0(x) = J_0(ix) = \sum_{k=0}^{\infty} \frac{\left(\frac{x}{2}\right)^{2k}}{k! \Gamma(k+1)},$$

где  $J_0(x)$  – это функция Бесселя нулевого порядка, а  $\Gamma(x)$  обозначает гамма-функцию.

Для  $x$  как вектор уравнение выше применяется к компонентам  $x$ .

#### Пример

`y=besseli0(1)` В о з в р а щ а е т 1.266.

#### См. также

`besselj()`, `bessely()`

**besselj()**

**Функция Бесселя n-го порядка.**

**Синтаксис**

```
jn=besselj(n,x)
```

**Аргументы**

Name	Type	Def. Range	Required
n	N	$[0, +\infty[$	✓
x	$\mathbb{R}, \mathbb{C}, \mathbb{R}^n, \mathbb{C}^n$	$] - \infty, +\infty[$	✓

**Описание**

Эта функция вычисляет функцию Бесселя n-го порядка действительного или комплексного числа или вектора.

$$J_n(x) = \sum_{k=0}^{\infty} \frac{(-1)^k \left(\frac{x}{2}\right)^{n+2k}}{k! \Gamma(n+k+1)},$$

где  $\Gamma(x)$  означает гамма-функцию.

Для x как вектор уравнение выше применяется к компонентам x.

**Пример**

```
y=besselj(1,1)  В о з в р а щ а е т  0,44.
```

**См. также**

```
besseli0(), bessely()
```

## bessely()

**Функция Бесселя второго рода n-го порядка.**

### Синтаксис

`yn=bessely(n,x)`

### Аргументы

Name	Type	Def. Range	Required
n	N	$[0, +\infty[$	✓
x	$\mathbb{R}, \mathbb{C}, \mathbb{R}^n, \mathbb{C}^n$	$] -\infty, +\infty[$	✓

### Описание

Эта функция вычисляет функцию Бесселя второго рода n-го порядка действительного или комплексного числа или вектора.

$$Y_n(x) = \lim_{m \rightarrow n} \frac{J_m(x) \cos m\pi - J_{-m}(x)}{\sin m\pi},$$

где  $J_m(x)$  означает функцию Бесселя перового рода n-го порядка.

Для x как вектор уравнение выше применяется к компонентам x.

### Пример

`y=bessely(1,1)` В о з в р а щ а е т -0.781.

### См. также

`besseli0()`, `besselj()`

## erf()

### Функция ошибки.

### Синтаксис

`y=erf (x)`

### Аргументы

Name	Type	Def. Range	Required
x	$\mathbb{R}, \mathbb{C}, \mathbb{R}^n, \mathbb{C}^n$	$] - \infty, + \infty [$	✓

### Описание

Эта функция осуществляет функцию ошибки для действительного или комплексного числа или вектора. Для

$$x \in \mathbb{R},$$

$$y = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt$$

Если  $x$  – комплексное число, обе, действительная и мнимая части становятся субъектом уравнения выше. Для  $x$  как вектор уравнение выше применяется к компонентам  $x$ .

### Пример

`y=erf(0.5)` В о з в р а щ а е т 0.520.

### См. также

`erfc()`, `erfinv()`, `erfcinv()`, `exp()`

**erfc()**

**Функция комплементарной ошибки.**

**Синтаксис**

```
y=erfc(x)
```

**Аргументы**

Name	Type	Def. Range	Required
x	$\mathbb{R}, \mathbb{C}, \mathbb{R}^n, \mathbb{C}^n$	$] -\infty, +\infty[$	✓

**Описание**

Эта функция вычисляет ошибку комплементарной функции действительного или комплексного числа или вектора. Для  $x \in \mathbb{R}$ ,

$$y = 1 - \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt$$

Если  $x$  – комплексное число, обе, действительная и мнимая, части становятся субъектом уравнения выше. Для  $x$  как вектор уравнение выше применяется к компонентам  $x$ .

**Пример**

```
y=erfc(0.5)  в о з в р а щ а е т  0.480.
```

**См. также**

```
erf(), erfinv(), erfcinv(), exp()
```

**erfinv()**

**Ошибка инверсной функции.**

**Синтаксис**

```
y=erfinv(x)
```

**Аргументы**

Name	Type	Def. Range	Required
x	$\mathbb{R}, \mathbb{C}, \mathbb{R}^n, \mathbb{C}^n$	$] - 1, + 1[$	✓

**Описание**

Эта функция осуществляет функцию ошибки инверсии для действительного или комплексного числа или вектора. Для  $-1 < x < 1$ ,

$$y = \operatorname{erf}^{-1}(x)$$

Если  $x$  – комплексное число, обе, действительная и мнимая, части становятся субъектом уравнения выше.

Для  $x$  как вектора уравнение выше применяется к компонентам  $x$ .

**Пример**

```
y=erfinv(0.8) В о з в р а щ а е т 0.906.
```

**См. также**

```
erf(), erfc(), erfcinv(), exp()
```

## erfcinv()

Ошибка инверсной комплементарной функции.

### Синтаксис

`y=erfcinv(x)`

### Аргументы

Name	Type	Def. Range	Required
x	$\mathbb{R}, \mathbb{C}, \mathbb{R}^n, \mathbb{C}^n$	$]0, +2[$	✓

### Описание

Эта функция осуществляет функцию ошибки комплементарной инверсии действительного или комплексного числа или вектора. Для  $0 < x < 2$ ,

$$y = \operatorname{erfc}^{-1}(x)$$

Если  $x$  – это комплексное число, обе, действительная и мнимая, части становятся субъектом уравнения выше.

Для  $x$  как вектор уравнение выше применяется к компонентам  $x$ .

### Пример

`y=erfcinv(0.5)` В о з в р а щ а е т 0.477.

### См. также

`erf()`, `erfc()`, `erfinv()`, `exp()`

**sinc()**

**Функция Sinc.**

**Синтаксис**

```
y=sinc(x)
```

**Аргументы**

Name	Type	Def. Range	Required
x	$\mathbb{R}, \mathbb{C}, \mathbb{R}^n, \mathbb{C}^n$	$]-\infty, +\infty[$	✓

**Описание**

Эта функция осуществляет sinc действительного или комплексного числа или вектора.

$$y = \begin{cases} \frac{\sin x}{x} & \text{for } x \neq 0 \\ 1 & \text{for } x = 0 \end{cases}$$

Для x как вектор уравнение выше применяется к компонентам x.

**Примеры**

```
y=sinc(-3) Возвращает 0.047,  
y=sinc(3+4*i) Возвращает -3.86-j3.86.
```

**См. также**

```
sin()
```

**step()**

**Функция Step.**

**Синтаксис**

```
y=step(x)
```

**Аргументы**

Name	Type	Def. Range	Required
x	$\mathbb{R}, \mathbb{C}, \mathbb{R}^n, \mathbb{C}^n$	$] -\infty, +\infty[$	✓

**Описание**

Эта функция вычисляет функцию step для действительного или комплексного числа или вектора. Для  $x \in \mathbb{R}$ ,

$$y = \begin{cases} 0 & \text{for } x < 0 \\ 0.5 & \text{for } x = 0 \\ 1 & \text{for } x > 0 \end{cases}$$

Если  $x$  – комплексное число, обе, действительная и мнимая, части становятся субъектом уравнения выше.

Для  $x$  как вектор уравнение выше применяется к компонентам  $x$ .

**Пример**

```
y=step(0.5)  В о з в р а щ а е т  1.
```

**См. также**

## 13.4.3 Анализ данных

### Базовая статистика

#### avg()

Среднее элементов вектора.

#### Синтаксис

`y=avg(x)`

#### Аргументы

Name	Type	Def. Range	Required
x	$\mathbf{R}, \mathbf{C}, \mathbf{R}^n, \mathbf{C}^n, \text{Range } xs : xe$	$] -\infty, +\infty[$	<input checked="" type="checkbox"/>

#### Описание

Эта функция возвращает сумму элементов действительного или комплексного вектора или области.

Для  $x \in \mathbb{C}^n$ :  $y = \frac{1}{n} \sum_{i=1}^n x_i, 1 \leq i \leq n$  (for vectors) or  $xs \leq i \leq xe$  (for ranges)

Для x действительного или комплексного числа возвращается сам x .

#### Пример

`y=avg(linspace(1,3,10))` В о з в р а щ а е т 2.

#### См. также

`sum()`, `max()`, `min()`

## cumavg()

**Кумулятивное среднее векторных элементов.**

### Синтаксис

`y=cumavg(x)`

### Аргументы

Name	Type	Def. Range	Required
x	$\mathbb{R}, \mathbb{C}, \mathbb{R}^n, \mathbb{C}^n$	$]-\infty, +\infty[$	✓

### Описание

Эта функция возвращает кумулятивное среднее элементов действительного или комплексного вектора.

Для  $x \in \mathbb{C}^n$ :  $y_k = \frac{1}{k} \sum_{i=1}^k x_i, 1 \leq k \leq n$

Для x действительного или комплексного числа возвращается сам x.

### Пример

`y=cumavg(linspace(1,3,3))` В о з в р а щ а е т 1, 1.5, 2.

### См. также

`cumsum()`, `cumprod()`, `avg()`, `sum()`, `prod()`, `max()`, `min()`

## max()

Максимальное значение.

### Синтаксис

$y = \max(x)$   
 $y = \max(a, b)$

### Аргументы

Name	Type	Def. Range	Required
x	$\mathbb{R}, \mathbb{C}, \mathbb{R}^n, \mathbb{C}^n, \text{Range } xs : xe$	$] -\infty, +\infty[$	✓
a	$\mathbb{R}, \mathbb{C}$	$] -\infty, +\infty[$	✓
b	$\mathbb{R}, \mathbb{C}$	$] -\infty, +\infty[$	✓

### Описание

Для первого синтаксиса ( $y = \max(x)$ ), эта функция возвращает максимальное значение действительного или комплексного числа или области.

For  $x \in \mathbb{R}^n$ :  $y = \max(x_i)$ ,  $1 \leq i \leq n$  (for vectors) or  $xs \leq i \leq xe$  (for ranges)

For  $x \in \mathbb{C}^n$ :  $y = \max(\pm |x_i|)$ ,  $1 \leq i \leq n$  (for vectors) or  $xs \leq i \leq xe$  (for ranges),

with sign  $\begin{cases} + & \text{for } |\arg(x_i)| \leq \frac{\pi}{2} \\ - & \text{else} \end{cases}$

Для  $x$  действительного или комплексного числа – это случай 1.

Второй синтаксис ( $y = \max(a, b)$ ) применяется, если два (обычно комплексные) числа  $a$  и  $b$  нуждаются в сравнении. В принципе, выбирается максимум абсолютного значения, но должно быть рассмотрено не находятся ли  $a$  и  $b$  в правой или левой половине комплексной области. Если последнее имеет место, отрицательное абсолютное значение  $a$  и  $b$  нуждаются во внимании (например, который относится к случаю отрицательного действительного числа), иначе берется положительное абсолютное значение:

$$y = \max(\pm |a|, \pm |b|),$$

$$\text{with } |a| \text{ sign } \begin{cases} + & \text{for } |\arg(a)| \leq \frac{\pi}{2} \\ - & \text{else} \end{cases} \text{ and } |b| \text{ sign } \begin{cases} + & \text{for } |\arg(b)| \leq \frac{\pi}{2} \\ - & \text{else} \end{cases}$$

### Пример

$y = \max(\text{linspace}(1, 3, 10))$  Возвращает 3.  
 $y = \max(1, 3)$  Возвращает 3.  
 $y = \max(1, 1+i)$  Возвращает  $1+j1$ .  
 $y = \max(1, -1+i)$  Возвращает 1.

### См. также

`min()`, `abs()`

## min()

### Minimum value.

#### Синтаксис

$y = \min(x)$   
 $y = \min(a, b)$

#### Аргументы

Name	Type	Def. Range	Required
x	$\mathbb{R}, \mathbb{C}, \mathbb{R}^n, \mathbb{C}^n, \text{Range } xs: xe$	$]-\infty, +\infty[$	✓
a	$\mathbb{R}, \mathbb{C}$	$]-\infty, +\infty[$	✓
b	$\mathbb{R}, \mathbb{C}$	$]-\infty, +\infty[$	✓

#### Описание

Для первого синтаксиса ( $y = \min(x)$ ) эта функция возвращает минимальное значение действительного или комплексного числа или области.

For  $x \in \mathbb{R}^n$ :  $y = \min(x_i), 1 \leq i \leq n$  (for vectors) or  $xs \leq i \leq xe$  (for ranges)

For  $x \in \mathbb{C}^n$ :  $y = \min(\pm |x_i|), 1 \leq i \leq n$  (for vectors) or  $xs \leq i \leq xe$  (for ranges),

with sign  $\begin{cases} + & \text{for } |\arg(x_i)| \leq \frac{\pi}{2} \\ - & \text{else} \end{cases}$

Для  $x$  действительного или комплексного числа – это случай 1.

Второй синтаксис ( $y = \min(a, b)$ ) находит применение, если два (обычно комплексных) числа  $a$  и  $b$  нуждаются в сравнении. В принципе, выбирается максимум абсолютных значений, но должно обращать внимание, будут ли  $a$  и  $b$  расположены в правой или левой половине комплексной области. Если последнее имеет место, отрицательное абсолютное значение  $a$  и  $b$  нуждаются во внимании (например, который относится к случаю отрицательного действительного числа), иначе берется положительное абсолютное значение:

$y = \max(\pm |a|, \pm |b|),$

with  $|a| \text{ sign } \begin{cases} + & \text{for } |\arg(a)| \leq \frac{\pi}{2} \\ - & \text{else} \end{cases}$  and  $|b| \text{ sign } \begin{cases} + & \text{for } |\arg(b)| \leq \frac{\pi}{2} \\ - & \text{else} \end{cases}$

#### Примеры

$y = \min(\text{linspace}(1, 3, 10))$  Возвращает 1.  
 $y = \min(1, 3)$  Возвращает 1.  
 $y = \min(1, 1+i)$  Возвращает 1.  
 $y = \min(1, -1+i)$  Возвращает  $-1+j1$ .

#### См. также

$\max()$ ,  $\text{abs}()$

**rms()**

**Корень из квадрата среднего векторных элементов.**

**Синтаксис**

```
y=rms(x)
```

**Аргументы**

Name	Type	Def. Range	Required
x	$\mathbb{R}, \mathbb{C}, \mathbb{R}^n, \mathbb{C}^n$	$] - \infty, + \infty [$	✓

**Описание**

Эта функция возвращает rms (корень квадратный из среднего) значение элементов действительного или комплексного вектора. Применение трапецеидального интегрального правила

Для  $x \in \mathbb{C}^n$ :  $y = \sqrt{\frac{1}{n} \sum_{i=1}^n a_i x_i x_i^*}$ ,  $1 \leq i \leq n$ ,  $a_i = \begin{cases} 1 & \text{for } 2 \leq i \leq n-1 \\ \frac{1}{2} & \text{for } i = 1 \text{ or } i = n \end{cases}$

Для x действительного или комплексного числа, возвращается сам |x|.

**Пример**

```
y=rms(linspace(1,2,8)) Возвращает 1.43.
```

**См. также**

```
variance(), stddev(), avg()
```

## runavg()

Теущее среднее векторных элементов.

### Синтаксис

`y=runavg(x,m)`

### Аргументы

Name	Type	Def. Range	Required
x	$\mathbb{R}, \mathbb{C}, \mathbb{R}^n, \mathbb{C}^n$	$] - \infty, + \infty[$	✓
m	N	$[1, + \infty[$	✓

### Описание

Эта функция возвращает скользящее среднее m элементов действительного или комплексного ветора.

Для

$$x \in \mathbb{C}^n: y_k = \frac{1}{m} \sum_{i=k}^{k+m-1} x_i, 1 \leq k \leq n$$

Для x действительного или комплексного числа, возвращается сам x.

### Пример

`y=runavg(linspace(1,3,6),2)` В о з в р а щ а е т 1.2, 1.6, 2, 2.4, 2.8.

### См. также

`cumavg()`, `cumsum()`, `avg()`, `sum()`

# stddev()

Стандартное отклонение элементов вектора.

## Синтаксис

```
y=stddev(x)
```

## Аргументы

Name	Type	Def. Range	Required
x	$\mathbb{R}, \mathbb{C}, \mathbb{R}^n, \mathbb{C}^n$	$] - \infty, + \infty [$	✓
m	N	$[1, + \infty [$	✓

## Описание

Эта функция возвращает stddev элементов действительного или комплексного вектора x.

Для  $x \in \mathbb{C}^n$ :  $y = \sqrt{\text{variance}(x)}$

Для x действительного или комплексного числа, возвращается 0.

## Пример

```
y=stddev(linspace(1,3,10)) В о з в р а щ а е т 0.673.
```

## См. также

```
stddev(), avg(), max(), min()
```

## variance()

Дисперсия векторных элементов.

### Синтаксис

`y=variance(x)`

### Аргументы

Name	Type	Def. Range	Required
x	$\mathbb{R}, \mathbb{C}, \mathbb{R}^n, \mathbb{C}^n$	$] - \infty, + \infty [$	✓

### Описание

Эта функция возвращает вариацию элементов действительного или комплексного вектора.

Для  $x \in \mathbb{C}^n$ :  $y = \frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})^2$ , where  $\bar{x}$  denotes mean (average) value of  $x$ .

Для  $x$  действительного или комплексного числа, возвращается 0.

### Пример

`y=variance(linspace(1,3,10))` В о з в р а щ а е т 0.453.

### См. также

`stddev()`, `avg()`, `max()`, `min()`

## Базовые операции

### **cumprod()**

**Кумулятивное умножение элементов вектора.**

#### **Синтаксис**

`y=cumprod(x)`

#### **Аргументы**

Name	Type	Def. Range	Required
x	$\mathbb{R}, \mathbb{C}, \mathbb{R}^n, \mathbb{C}^n$	$] -\infty, +\infty[$	✓

#### **Описание**

Эта функция возвращает кумулятивное умножение элементов действительного или комплексного вектора.

Для  $x \in \mathbb{C}^n$ :  $y_k = \prod_{i=1}^k x_i, 1 \leq k \leq n$

Для x как действительное или комплексное число, возвращается сам x.

#### **Пример**

`y=cumprod(linspace(1,3,3))` В о з в р а щ а е т 1, 2, 6.

#### **См. также**

`cumsum()`, `cumavg()`, `prod()`, `sum()`, `avg()`, `max()`, `min()`

# cumsum()

Кумулятивная сумма элементов вектора.

## Синтаксис

```
y=cumsum(x)
```

## Аргументы

Name	Type	Def. Range	Required
x	$\mathbb{R}, \mathbb{C}, \mathbb{R}^n, \mathbb{C}^n$	$] - \infty, + \infty [$	✓

## Описание

Эта функция возвращает кумулятивную сумму элементов действительного или комплексного вектора.

Для  $x \in \mathbb{C}^n$ :  $y_k = \sum_{i=1}^k x_i, 1 \leq k \leq n$

Для x как действительное или комплексное число, возвращается сам x.

## Пример

```
y=cumsum(linspace(1,3,3))
```

 Возвращает 1, 3, 6.

## См. также

```
cumprod(), cumavg(), sum(), prod(), avg(), max(), min()
```

## interpolate()

Эквидистантная сплайн-интерполяция вектора данных.

### Синтаксис

```
z=interpolate(y,t,m)  
z=interpolate(y,t)
```

### Аргументы

Name	Type	Def. Range	Required	Default
y	$\mathbb{R}^n, \mathbb{C}^n$	$]-\infty, +\infty[$	✓	
t	$\mathbb{R}^n, \mathbb{C}^n$	$]-\infty, +\infty[$	✓	
m	N	$[3, +\infty[$		64

### Описание

Эта функция использует сплайн-интерполяцию для интерполяции между точками вектора  $y(t)$ .  
Если число проб  $n$  не задано, принимается значение по умолчанию  $n = 64$ .

### Пример

```
z=interpolate(linspace(0,2,3)*linspace(0,2,3),linspace(0,2,3))
```

возвращает гладкую параболическую кривую:

Используйте декартову систему (Cartesian diagram) для ее отображения.

### См. также

`sum()`, `prod()`

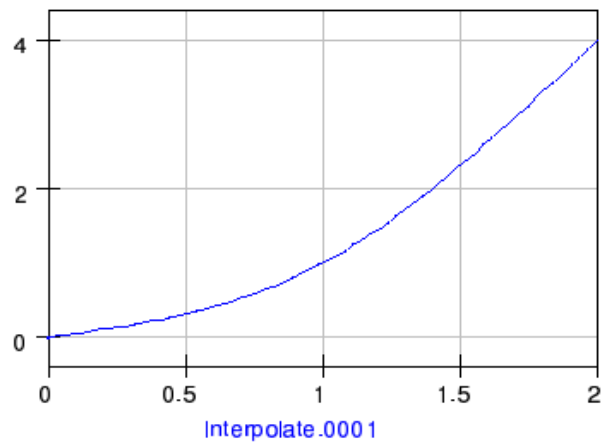


Рис. 13.4. Кривая интерполяции

# prod()

Умножение элементов вектора.

## Синтаксис

```
y=prod(x)
```

## Аргументы

Name	Type	Def. Range	Required
x	$\mathbb{R}, \mathbb{C}, \mathbb{R}^n, \mathbb{C}^n$	$] - \infty, + \infty [$	✓

## Описание

Эта функция возвращает результат умножения элементов действительного или комплексного вектора.

Для  $x \in \mathbb{C}^n$ :  $y = \prod_{i=1}^n x_i$

Для x как действительное или комплексное число, возвращается x.

## Пример

```
y=prod(linspace(1,3,10)) В о з в р а щ а е т 583.
```

## См. также

```
sum(), avg(), max(), min()
```

**sum()**

**Сумма элементов вектора.**

**Синтаксис**

```
y=sum(x)
```

**Аргументы**

Name	Type	Def. Range	Required
x	$\mathbb{R}, \mathbb{C}, \mathbb{R}^n, \mathbb{C}^n$	$] - \infty, + \infty[$	✓

**Описание**

Эта функция возвращает сумму элементов действительного или комплексного вектора.

Для  $x \in \mathbb{C}^n$ :  $y = \sum_{i=1}^n x_i$

Для x как действительное или комплексное число, возвращается x.

**Пример**

```
y=sum(linspace(1,3,10)) Возвращает 20.
```

**См. также**

```
prod(), avg(), max(), min()
```

# xvalue()

Возвращает x-значение, которое ассоциировано с y-значением, ближайшим к заданному y-значению заданного вектора.

## Синтаксис

```
x=xvalue(f,yval)
```

## Аргументы

Name	Type	Def. Range	Required
f	$\mathbb{R}^n, \mathbb{C}^n$	$] - \infty, + \infty[$	✓
yval	$\mathbb{R}, \mathbb{C}$	$] - \infty, + \infty[$	✓

## Описание

Эта функция возвращает x-значение, ассоциированное с y-значением, ближайшим к yval в заданном векторе f. Следовательно, вектор f должен иметь однозначную зависимость от данных.

## Пример

```
x=xvalue(f,1).
```

## См. также

```
yvalue(), interpolate()
```

# yvalue()

Возвращает у-значение данного вектора, локализованного возле заданного х-значения.

## Синтаксис

```
y=yvalue(f,xval)
```

## Аргументы

Name	Type	Def. Range	Required
f	$\mathbb{R}^n, \mathbb{C}^n$	$]-\infty, +\infty[$	$\checkmark$
xval	$\mathbb{R}, \mathbb{C}$	$]-\infty, +\infty[$	$\checkmark$

## Описание

Эта функция возвращает у-значение заданного вектора f, который локализован вблизи х-значения xval. Следовательно, вектор f должен иметь однозначную зависимость данных.

## Пример

```
y=yvalue(f,1).
```

## См. также

```
xvalue(), interpolate()
```

## Дифференцирование и интегрирование

### diff()

Производная вектора относительно другого вектора.

#### Синтаксис

`z=diff(y,x,n)`

#### Аргументы

Name	Type	Def. Range	Required	Default
y	$\mathbb{R}^k, \mathbb{C}^k$	$]-\infty, +\infty[$	✓	
x	$\mathbb{R}^m, \mathbb{C}^m$	$]-\infty, +\infty[$	✓	
n	N			1

#### Описание

Эта функция дает численное дифференцирование вектора y по отношению к вектору x. Если дополнительный целый параметр n задан, n-ая производная вычисляется. Дифференцирование выполняется для  $N=\min(k,m)$  элементов. Для n=1,

$$\frac{\Delta y_i}{\Delta x_i} = \begin{cases} \frac{1}{2} \left( \frac{y_i - y_{i-1}}{x_i - x_{i-1}} + \frac{y_{i+1} - y_i}{x_{i+1} - x_i} \right) & \text{for } N-1 > i > 0 \\ \frac{y_{i+1} - y_i}{x_{i+1} - x_i} & \text{for } i = 0 \\ \frac{y_i - y_{i-1}}{x_i - x_{i-1}} & \text{for } i = N-1 \end{cases}$$

Если n>1, результат дифференцирования выше применяется к y и вышеупомянутый шаг дифференцирования повторяется до тех пор, пока число шагов не станет равно n.

#### Пример

`z=diff(linspace(1,3,3),linspace(2,3,3))` Возвращает 2, 2, 2.

#### См. также

`integrate()`, `sum()`, `max()`, `min()`

# integrate()

## Интегрирование вектора.

### Синтаксис

```
z=integrate(y,h)
```

### Аргументы

Name	Type	Def. Range	Required
y	$\mathbb{R}, \mathbb{C}, \mathbb{R}^n, \mathbb{C}^n$	$] - \infty, + \infty[$	✓
h	$\mathbb{R}, \mathbb{C}$	$] - \infty, + \infty[$	✓

### Описание

Эта функция численно интегрирует вектор x относительно дифференциала h. Метод интеграции согласно правилу трапеции:

$$\int f(t) dt \approx h \left( \frac{y_0}{2} + y_1 + y_2 + \dots + y_{n-1} + \frac{y_n}{2} \right)$$

### Пример

Вычисление приближения интеграла  $\int_1^3 t dt$  с использованием 105 точек:

```
z=integrate(linspace(1,3,105)*linspace(1,3,105),0.02)  в о з в р а щ а е т  4.
```

### См. также

```
diff(), sum(), max(), min()
```

## Обработка сигналов

### dft()

Дискретное преобразование Фурье.

#### Синтаксис

`y=dft(v)`

#### Аргументы

Name	Type	Def. Range	Required
v	$\mathbb{R}^n, \mathbb{C}^n$	$]-\infty, +\infty[$	✓

#### Описание

Эта функция рассчитывает дискретное преобразование Фурье вектора `v`. Преимущество этой функции над `fft()` в том, что число `n` компонент `v` произвольно, и в последней `n` должно быть степени 2. Недостаток в том, что `dft()` медленнее и менее точна, чем `fft()`.

#### Пример

Вычисление спектра у DC сигнала:

<code>y=dft(linspace(1,1,7))</code> returns	y
	1
	-1.59e-17+j1.59e-17
	⋮
	2.22e-16-j1.11e-16

Пожалуйста заметьте, что в этом примере используется 7 точек меток времени для вектора `v`. Поскольку 7 не степень 2, это же выражение, использованное совместно с функцией `fft()`, приведет к неверным результатам. Заметьте также ошибку округления, где “0” будет корректным значением.

#### См. также

`idft()`, `fft()`, `ifft()`, `Freq2Time()`, `Time2Freq()`

### fft()

Быстрое преобразование Фурье.

#### Синтаксис

`y=fft(v)`

#### Аргументы

Name	Type	Def. Range	Required
$v$	$\mathbb{R}^n, \mathbb{C}^n$	$] - \infty, + \infty[$	$\checkmark$

Описание

Эта функция рассчитывает быстрое преобразование Фурье (Fast Fourier Transform – FFT) вектора  $v$ . Число  $n$  компонент  $v$  должно быть сепени 2.

Пример

Вычисление спектра у DC сигнала:

`y=fft(linspace(1,1,8))`
returns

y

1

0

⋮

0

См. также

`ifft(), dft(), idft(), Freq2Time(), Time2Freq()`

# idft()

Инверсное дискретное преобразование Фурье.

## Синтаксис

```
y=idft(v)
```

## Аргументы

Name	Type	Def. Range	Required
v	$\mathbb{R}^n, \mathbb{C}^n$	$]-\infty, +\infty[$	✓

## Описание

Эта функция рассчитывает инверсное дискретное преобразование Фурье (Inverse Discrete Fourier Transform – IDFT) вектора v. Преимущество этой функции по сравнению с ifft() в том, число n компонент v произвольно, тогда как для последней должно быть степенью 2. Недостаток в том, что idft() медленнее и менее точна, чем ifft().

## Пример

Вычисление времен функции y, принадлежащих к белому спектру:

```
y=idft(linspace(1,1,7)) returns
```

y
7
-1.11e-16-j1.11e-16
⋮
1.55e-15+j7.77e-16

Пожалуйста заметьте, что в этом примере используется 7 точек для спектра вектора v. Поскольку 7 не степень 2, то же самое выражение, использованное вместе с функцией ifft() даст неверные результаты. Заметьте, что ошибка округления “0” будет правильным значением.

## См. также

```
dft(), ifft(), fft(), Freq2Time(), Time2Freq()
```

# ifft()

Инверсное быстрое преобразование Фурье.

## Синтаксис

```
y=ifft(v)
```

## Аргументы

Name	Type	Def. Range	Required
$v$	$\mathbb{R}^n, \mathbb{C}^n$	$]-\infty, +\infty[$	$\checkmark$

## Описание

Эта функция рассчитывает инверсное быстрое преобразование Фурье (Inverse Fast Fourier Transform – IFFT) вектора  $v$ . Число  $n$  компонент  $v$  должно быть степени 2.

## Пример

Вычисление времен функции  $y$ , принадлежащих белому спектру:

```
y=ifft(linspace(1,1,8)) returns
```

y
8
0
⋮
0

## См. также

`fft()`, `dft()`, `idft()`, `Freq2Time()`, `Time2Freq()`

## Time2Freq()

Интерпретированное дискретное преобразование Фурье.

### Синтаксис

`y=Time2Freq(v,t)`

### Аргументы

Name	Type	Def. Range	Required
v	$\mathbb{R}^n, \mathbb{C}^n$	$]-\infty, +\infty[$	✓
t	$\mathbb{R}^k, \mathbb{C}^k$	$]-\infty, +\infty[$	✓

### Описание

Эта функция рассчитывает дискретное преобразование Фурье (DFT) вектора v с рассмотрением вектора времени t.

### Пример

Вычисление спектра y(f) DC сигнала:

`y=Time2Freq(linspace(1,1,7),linspace(0,1,2))` В о з в р а щ а е т

Frequency	y
0	1
0.167	-1.59e-17+j1.59e-17
⋮	⋮
1	2.22e-16-j1.11e-16

Пожалуйста заметьте, что для времени вектора v использовано 7 точек. Заметьте также, что ошибка округления при  $t>0$  равная “0” будет правильным значением.

### См. также

`idft()`, `fft()`, `ifft()`, `Freq2Time()`

# Freq2Time()

Интерпретированное инверсное дискретное преобразование Фурье.

## Синтаксис

```
y=Freq2Time(v,f)
```

## Аргументы

Name	Type	Def. Range	Required
v	$\mathbb{R}^n, \mathbb{C}^n$	$]-\infty, +\infty[$	✓
f	$\mathbb{R}^k, \mathbb{C}^k$	$]-\infty, +\infty[$	✓

## Описание

Эта функция рассчитывает инверсное дискретное преобразование Фурье (IDFT) вектора v с рассмотрением частоты вектора f.

## Пример

Вычисление времен функции y(t), принадлежащих белому спектру:

```
y=Freq2Time(linspace(1,1,7),linspace(0,1,2)) В о з в р а щ а е т
```

Frequency	y
0	7
0.167	-1.11e-16-j1.11e-16
⋮	⋮
1	1.55e-15+j7.77e-16

Пожалуйста заметьте, что в этом примере для спектра вектора v использовано 7 точек. Заметьте также, что ошибки округления при t>0 равные “0” будут правильными значениями.

## См. также

```
dft(), ifft(), fft(), Time2Freq()
```

**kbd()**

**Кайзера-Бесселя производное окно.**

**Синтаксис**

```
y=kbd(a,n)
y=kbd(a)
```

**Аргументы**

Name	Type	Def. Range	Required	Default
a	<b>R</b>	$] - \infty, + \infty[$	$\sqrt{\phantom{x}}$	
n	<b>N</b>	$[1, + \infty[$		64

**Описание**

Эта функция генерирует окно Кайзера-Бесселя согласно

$$y_k = \frac{\sum_{i=0}^k I_0 \left( \pi a \sqrt{1 - \left(\frac{4i}{n} - 1\right)} \right)}{\sum_{i=0}^{\frac{n}{2}} I_0 \left( \pi a \sqrt{1 - \left(\frac{4i}{n} - 1\right)} \right)},$$
$$y_{n-k-1} = y_k$$
$$\text{for } 0 \leq k < \frac{n}{2}$$

Если параметр n не задан, применяется n=64.

**Пример**

```
y=kbd(0.1,4) В о з в р а щ а е т .
```

**См. также**

```
dft(), ifft(), fft()
```

## 13.5 Электронные функции

### 13.5.1 Конвертирование единиц

dB()

dB значение.

Синтаксис

$y = \text{dB}(x)$

Аргументы

Name	Type	Def. Range	Required
x	$\mathbb{R}, \mathbb{C}, \mathbb{R}^n, \mathbb{C}^n$	$] -\infty, +\infty[$	✓

Описание

Эта функция возвращает dB значение действительного или комплексного числа или вектора.

$$y = 20 \log |x|$$

Для x как вектор уравнение выше применяется к компонентам x.

Пример

$y = \text{db}(10)$  возвращает 20.

См. также

$\log_{10}()$

## dbm()

Конвертирует напряжение в мощность в dBm.

### Синтаксис

$y = \text{dBm}(u, Z_0)$

$y = \text{dBm}(u)$

### Аргументы

Name	Type	Def. Range	Required	Default
$u$	$\mathbb{R}, \mathbb{C}, \mathbb{R}^n, \mathbb{C}^n$	$]-\infty, +\infty[$	✓	
$Z_0$	$\mathbb{R}, \mathbb{C}, \mathbb{R}^n, \mathbb{C}^n$	$]-\infty, +\infty[$		50

### Описание

Эта функция возвращает соответствующую dBm мощность действительного или комплексного напряжения или вектора  $u$ . Импеданс  $Z_0$  относится либо к заданному, либо 50  $\Omega$ .

$$y = 10 \log \frac{|u|^2}{Z_0 0.001W}$$

Для  $u$  как вектор уравнение выше применяется к компонентам  $u$ .

Пожалуйста заметьте, что  $u$  рассматривается как rms значение, а не амплитуда.

### Пример

$y = \text{dbm}(1)$  возвращает 13.

### См. также

`dbm2w()`, `w2dbm()`, `log10()`

**dbm2w()**

**Конвертирует мощность в dBm в мощность в Watts.**

**Синтаксис**

```
y=dbm2w( x )
```

**Аргументы**

Name	Type	Def. Range	Required
x	$\mathbb{R}, \mathbb{C}, \mathbb{R}^n, \mathbb{C}^n$	$] - \infty, + \infty [$	✓

**Описание**

Эта функция конвертирует действительную или комплексную мощность или вектор мощности, заданные в dBm, в соответствующую мощность в Watts.

$$y = 0.001 \cdot 10^{\frac{x}{10}}$$

Для x как вектор уравнение выше применяется к компонентам x.

**Пример**

```
y=dbm2w(10)  В о з в р а щ а е т  0.01.
```

**См. также**

```
dbm( ),  w2dbm( )
```

# w2dbm()

Конвертирует мощность в Watts в мощность в dBm.

## Синтаксис

y=w2dBm(x)

## Аргументы

Name	Type	Def. Range	Required
x	$\mathbb{R}, \mathbb{C}, \mathbb{R}^n, \mathbb{C}^n$	$] - \infty, + \infty [$	✓

## Описание

Эта функция конвертирует действительную или комплексную мощность или вектор мощности, заданные в Watts, в соответствующую мощность в dBm.

$$y = 10 \log \frac{x}{0.001W}$$

Для x как вектор уравнение выше применяется к компонентам x.

## Пример

y=w2dbm(1) В о з в р а щ а е т 30.

## См. также

dbm(), dbm2w(), log10()

## 13.5.2 Коэффициент отражения и VSWR

### rtoswr()

Конвертирует коэффициент отражения в отношение напряжения стоячей волны (voltage standing wave ratio – VSWR).

#### Синтаксис

```
s=rtoswr(r)
```

#### Аргументы

Name	Type	Def. Range	Required
r	$\mathbb{R}, \mathbb{C}, \mathbb{R}^n, \mathbb{C}^n$	$ r  \leq 1$	✓

#### Описание

Для действительного или комплексного коэффициента отражения r эта функция вычисляет соответствующее отношение напряжения стоячей волны (VSWR) s согласно

$$s = \frac{1 + |r|}{1 - |r|}$$

VSWR – это действительное число, и обычно задается в написании “s : 1”.

Для r как вектор уравнение выше применяется к компонентам r.

#### Примеры

```
s=rtoswr(0) Возвращает 1.  
s=rtoswr(0.1+0.2*i) Возвращает 1.58.
```

#### См. также

```
ytor(), ztor(), rtoy(), rtoz()
```

**rtoz()**

**Конвертирует коэффициент отражения в полную проводимость.**

**Синтаксис**

```
y=rtoz(r)
y=rtoz(r, Z0)
```

**Аргументы**

Name	Type	Def. Range	Required	Default
r	$\mathbb{R}, \mathbb{C}, \mathbb{R}^n, \mathbb{C}^n$	$ r  \leq 1$	✓	
Z0	$\mathbb{R}, \mathbb{C}$	$] -\infty, +\infty[$		50

**Описание**

Для действительного или комплексного коэффициента отражения r эта функция вычисляет соответствующую полную проводимость y согласно

$$y = \frac{1}{Z_0} \frac{1 - r}{1 + r}$$

Если указываемый импеданс не предоставлен, функция принимает Z0 = 50Ω.

Для r как вектор уравнение выше применяется к компонентам r.

**Пример**

```
y=rtoz(0.333) В о з в р а щ а е т  0.01.
```

**См. также**

```
ytor(), ztor(), rtoswr()
```

**rtoz()**

**Конвертирует коэффициент отражения в импеданс.**

**Синтаксис**

```
z=rtoz(r)
z=rtoz(r, Z0)
```

**Аргументы**

Name	Type	Def. Range	Required	Default
r	$\mathbb{R}, \mathbb{C}, \mathbb{R}^n, \mathbb{C}^n$	$ r  \leq 1$	✓	
Z0	$\mathbb{R}, \mathbb{C}$	$] -\infty, +\infty[$		50

**Описание**

Для действительного или комплексного коэффициента отражения r эта функция вычисляет соответствующий импеданс Z согласно

$$Z = Z_0 \frac{1 - r}{1 + r}$$

Если указанный импеданс Z0 не предоставлен, функция принимает  $Z_0 = 50\Omega$ .

Для r как вектор уравнение выше применяется к компонентам r.

**Пример**

```
z=rtoz(0.333)  В о з в р а щ а е т  99.9.
```

**См. также**

```
ztor(), ytor(), rtoswr()
```

# ytor()

Конвертирует полную проводимость в коэффициент отражения.

## Синтаксис

```

r=ytor(Y)
r=ytor(Y, Z0)

```

## Аргументы

Name	Type	Def. Range	Required	Default
Y	$\mathbb{R}, \mathbb{C}, \mathbb{R}^n, \mathbb{C}^n$	$] - \infty, + \infty[$	✓	
Z0	$\mathbb{R}, \mathbb{C}$	$] - \infty, + \infty[$		50

## Описание

Для действительной или комплексной полной проводимости у эта функция вычисляет соответствующий коэффициент отражения согласно

$$r = \frac{1 - Y Z_0}{1 + Y Z_0}$$

Для Y как вектора уравнение выше применяется к компонентам Y.

Если указанный импеданс Z0 не предоставлен, функция принимает  $Z_0 = 50 \Omega$ .

Часто dB значение задается для коэффициента отражения, так называемые потери на отражение “return loss”:

$$RL = -20 \log |r| \text{ [dB]}$$

## Пример

```

r=ytor(0.01)  В о з в р а щ а е т  0.333.

```

## См. также

```

rtoy(), rtoz(), rtoswr(), log10(), dB()

```

# ztor()

Конвертирует импеданс в коэффициент отражения.

## Синтаксис

```
r=ztor(Z)
r=ztor(Z, Z0)
```

## Аргументы

Name	Type	Def. Range	Required	Default
Z	$\mathbb{R}, \mathbb{C}, \mathbb{R}^n, \mathbb{C}^n$	$] - \infty, + \infty[$	✓	
Z0	$\mathbb{R}, \mathbb{C}$	$] - \infty, + \infty[$		50

## Описание

Для действительного или комплексного импеданса Z эта функция вычисляет соответствующий коэффициент отражения согласно

$$r = \frac{Z - Z_0}{Z + Z_0}$$

Для Z как вектор уравнение выше применяется к компонентам Z.  
Если указанный импеданс Z0 не предоставлен, функция принимает  $Z_0 = 50 \Omega$ .  
Часто dB значение задается для коэффициента отражения, так называемые потери на отражение “return loss”:

$$RL = -20 \log |r| \text{ [dB]}$$

## Пример

```
r=ztor(100)  В о з в р а щ а е т  0.333.
```

## См. также

```
rtoz(), rtoy(), rtoswr(), log10(), dB()
```

## 13.5.3 N-портовые матричные преобразования

### stos()

Конвертирует матрицу S-параметров в матрицу S-параметров с разным ссылочным импедансом(ми).

#### Синтаксис

```
y=stos(S, Zref)
y=stos(S, Zref, Z0)
```

#### Аргументы

Name	Type	Def. Range	Required	Default
S	$\mathbb{R}^{n \times n}, \mathbb{C}^{n \times n}$	$ S_{ij}  \in ]-\infty, +\infty[, 1 \leq i, j \leq n$ $ S_{ii}  \leq 1, 1 \leq i \leq n$	✓	
Zref	$\mathbb{R}, \mathbb{C}, \mathbb{R}^n, \mathbb{C}^n$	$]-\infty, +\infty[$	✓	
Z0	$\mathbb{R}, \mathbb{C}, \mathbb{R}^n, \mathbb{C}^n$	$]-\infty, +\infty[$		50

#### Описание

Эта функция конвертирует действительные или комплексные параметры матрицы S в матрицу Y. S имеет ссылаемый импеданс Zref, хотя создаваемая дисперсная матрица Y имеет ссылаемый импеданс Z0.

Если указанный импеданс Z0 не предоставлен, функция принимает  $Z0 = 50\Omega$ .

Оба, Zref и Z0, могут быть действительным или комплексным числом или вектором. В последнем случае функция оперирует на элементах Zref и Z0.

#### Пример

Преобразование нагруженных на 50Ω S-параметров в нагруженные на 100Ω S-параметры:

```
S2=stos(eye(2)*0.1,50,100) В о з в р а щ а е т
```

-0.241	0
0	-0.241

#### См. также

twoport(), stoy(), stoz()

### stoy()

Конвертирует матрицу S-параметров в матрицу Y-параметров.

#### Синтаксис

```
Y=stoy(S)
Y=stoy(S, Zref)
```

#### Аргументы

Name	Type	Def. Range	Required	Default
S	$\mathbb{R}^{n \times n}, \mathbb{C}^{n \times n}$	$ S_{ij}  \in ]-\infty, +\infty[, 1 \leq i, j \leq n$ $ S_{ii}  \leq 1, 1 \leq i \leq n$	✓	
Zref	$\mathbb{R}, \mathbb{C}, \mathbb{R}^n, \mathbb{C}^n$	$]-\infty, +\infty[$		50

Описание

Эта функция конвертирует действительную или комплексную матрицу параметров S в матрицу полной проводимости Y. S имеет ссылочный импеданс Zref, который принимается Zref = 50Ω, если не предоставлен пользователем.

Zref может быть действительным или комплексным числом или вектором. В последнем случае функция оперирует с элементами Zref.

Пример

Y=stoy(eye(2)\*0.1,100) возвращает

0.00818	0
0	0.00818

См. также

twoport(), stos(), stoz(), ytos()

stoz()

Конвертирует матрицу S-параметров в матрицу Z-параметров.

Синтаксис

```
Z=stoz(S)
Z=stoz(S, Zref)
```

Аргументы

Name	Type	Def. Range	Required	Default
S	$\mathbb{R}^{n \times n}, \mathbb{C}^{n \times n}$	$ S_{ij}  \in ]-\infty, +\infty[, 1 \leq i, j \leq n$ $ S_{ii}  \leq 1, 1 \leq i \leq n$	✓	
Zref	$\mathbb{R}, \mathbb{C}, \mathbb{R}^n, \mathbb{C}^n$	$]-\infty, +\infty[$		50

Описание

Эта функция конвертирует действительные или комплексные параметры матрицы S в матрицу импеданса Z. S имеет ссылочный импеданс *Zref*, который принимается *Zref* = 50Ω, если не задан пользователем.

*Zref* может быть действительным или комплексным числом или вектором. В последнем случае функция оперирует с элементами *Zref*.

Пример

```
Z=stoz(eye(2)*0.1,100) В о з в р а щ а е т
```

122	0
0	122

См. также

```
twoport(), stos(), stoy(), ztos()
```

**twoport()**

Конвертирует двух-портовую матрицу из одного представления в другое.

**Синтаксис**

```
U=twoport(X, from, to)
```

**Аргументы**

Name	Type	Def. Range	Required
X	$\mathbb{R}^{2 \times 2}, \mathbb{C}^{2 \times 2}$	$]-\infty, +\infty[$	✓
from	Character	$\{'Y', 'Z', 'H', 'G', 'A', 'S', 'T'\}$	✓
to	Character	$\{'Y', 'Z', 'H', 'G', 'A', 'S', 'T'\}$	✓

**Описание**

Функция конвертирует действительную или комплексную двух-портовую матрицу *X* из одного представления в другое.

**Пример**

Преобразование двух-портовой Y матрицы Y1 в матрицу Z:

```
Y1=eye(2)*0.1
```

```
Z1=twoport(Y1,'Y','Z') В о з в р а щ а е т
```

10	0
0	10

**См. также**

```
stos(), ytos(), ztos(), stoz(), stoy(), ytoz(), ztoy()
```

**ytos()**

Конвертирует матрицу Y-параметров в матрицу S-параметров.

**Синтаксис**

```
S=ytos(Y)
S=ytos(Y, Z0)
```

**Аргументы**

Name	Type	Def. Range	Required	Default
Y	$\mathbb{R}^{n \times n}, \mathbb{C}^{n \times n}$	$] - \infty, + \infty[$	✓	
Z0	$\mathbb{R}, \mathbb{C}, \mathbb{R}^n, \mathbb{C}^n$	$] - \infty, + \infty[$		50

**Описание**

Эта функция конвертирует действительную или комплексную матрицу проводимости Y в дисперсную матрицу S. Y имеет ссылочный импеданс Z0, который принимается Z0 = 50Ω, если не предоставлен пользователем.

Z0 может быть действительным или комплексным числом или вектором. В последнем случае функция оперирует с элементами Z0.

**Пример**

```
S=ytos(eye(2)*0.1,100) В о з в р а щ а е т
```

-0.818	0
0	-0.818

**См. также**

```
twoport(), stos(), ztos(), stoy()
```

ytoz()

Конвертирует матрицу Y-параметров в матрицу Z-параметров.

Синтаксис

Z=ytoz(Y)

Аргументы

Name	Type	Def. Range	Required
Y	$\mathbb{R}^{n \times n}, \mathbb{C}^{n \times n}$	$]-\infty, +\infty[$	✓

Описание

Эта функция конвертирует действительную или комплексную матрицу полной проводимости Y в матрицу импеданса Z.

Пример

Z=ytoz(eye(2)\*0.1)    В о з в р а щ а е т

10	0
0	10

См. также

twoport(), ztoy()

ztos()

Конвертирует матрицу Z-параметров в матрицу S-параметров.

Синтаксис

```
S=ztos(Z)
S=ztos(Z, Z0)
```

Аргументы

Name	Type	Def. Range	Required	Default
Z	$\mathbb{R}^{n \times n}, \mathbb{C}^{n \times n}$	$]-\infty, +\infty[$	✓	
Z0	$\mathbb{R}, \mathbb{C}, \mathbb{R}^n, \mathbb{C}^n$	$]-\infty, +\infty[$		50

Описание

Эта функция конвертирует действительную или комплексную матрицу импедансов Z в матрицу S. Z имеет ссылочный импеданс Z0, который принимается Z0 = 50Ω, если не предоставлен пользователем.

Z0 может быть действительным или комплексным числом или вектором. В последнем случае функция оперирует с элементами Z0.

Пример

```
S=ztos(eye(2)*0.1,100) В о з в р а щ а е т
```

-0.998	0
0	-0.998

См. также

```
twoport(), twoport(), stos(), ytos(), stoz()
```

ztoy()

Конвертирует матрицу Z-параметров в матрицу Y-параметров.

Синтаксис

```
Y=ztoy(Z)
```

Аргументы

Name	Type	Def. Range	Required
Z	$\mathbb{R}^{n \times n}, \mathbb{C}^{n \times n}$	$]-\infty, +\infty[$	✓

Описание

Эта функция конвертирует действительную или комплексную матрицу импедансов Z в матрицу полной проводимости Y.

Пример

```
Y=ztoy(eye(2)*0.1) В о з в р а щ а е т
```

10	0
0	10

См. также

```
twoport(), ytoz()
```

## 13.5.4 Усилители

### GaCircle()

Окружность(и) с постоянной допустимой мощностью усиления Ga в плоскости источника.

#### Синтаксис

```
y=GaCircle(X,Ga,v)
y=GaCircle(X,Ga,n)
y=GaCircle(X,Ga)
```

#### Аргументы

Name	Type	Def. Range	Required	Default
X	$\mathbb{R}^{2 \times 2 \times p}, \mathbb{C}^{2 \times 2 \times p}$	$]-\infty, +\infty[$	✓	
v	$\mathbb{R}^n$	$[0, 360]^p$		
Ga	$\mathbb{R}, \mathbb{R}^m$	$[0, +\infty[$	✓	
n	N	$[2, +\infty[$		64

#### Описание

Эта функция генерирует точки окружности постоянной допустимой мощности усиления GA в комплексной плоскости источника (rS) усилителя. Усилитель описывается двух-портовой матрицей S-параметров – S. Радиус r и центр c этой окружности вычисляется следующим образом:

$$r = \frac{\sqrt{1 - 2 \cdot K \cdot g_A \cdot |S_{12}S_{21}| + g_A^2 \cdot |S_{12}S_{21}|^2}}{|1 + g_A \cdot (|S_{11}|^2 - |\Delta|^2)|} \text{ and } c = \frac{g_A (S_{11}^* - S_{22} \Delta^*)}{1 + g_A (|S_{11}|^2 - |\Delta|^2)},$$

где  $g_A = \frac{G_A}{|S_{21}|^2}$  а K – Rollet фактор стабильности.  $\Delta$  означает определитель S.

Точки окружности могут быть заданы углом вектора v, где угол должен быть в градусах. Другая возможность – это специфицировать число n угловых равноудаленных точек окружности. Если не задан дополнительный аргумент X, берется 64 точки. Допустимое усиление мощности может быть также специфицировано вектором Ga, приводя к генерации m окружностей, где m – это величина Ga.

Пожалуйста, обратитесь также к “Qucs -Technical Papers”, глава 1.5.

#### Пример

```
v=GaCircle(S)
```

#### См. также

```
GpCircle(), Rollet()
```

## GpCircle()

Окружность(и) с постоянным рабочим усилением мощности **Gp** в плоскости нагрузки.

### Синтаксис

```
y=GpCircle(X,Gp,v)
y=GpCircle(X,Gp,n)
y=GpCircle(X,Gp)
```

### Аргументы

Name	Type	Def. Range	Required	Default
X	$\mathbb{R}^{2 \times 2 \times p}, \mathbb{C}^{2 \times 2 \times p}$	$] -\infty, +\infty[$	✓	
v	$\mathbb{R}^n$	$[0, 360]^p$		
Gp	$\mathbb{R}, \mathbb{R}^m$	$[0, +\infty[$	✓	
n	N	$[2, +\infty[$		64

### Описание

Эта функция генерирует точки окружности постоянного рабочего усиления мощности **Gp** в комплексной плоскости нагрузки (**rL**) усилителя. Усилитель описывается двух-портовой матрицей S-параметров – **S**. Радиус **r** и центр **c** этой окружности вычисляется следующим образом:

$$r = \frac{\sqrt{1 - 2 \cdot K \cdot g_P \cdot |S_{12}S_{21}| + g_P^2 \cdot |S_{12}S_{21}|^2}}{|1 + g_P \cdot (|S_{22}|^2 - |\Delta|^2)|} \text{ and } c = \frac{g_A (S_{22}^* - S_{11} \Delta^*)}{1 + g_P (|S_{22}|^2 - |\Delta|^2)},$$

где  $g_A = \frac{G_P}{|S_{21}|^2}$  а **K** – Rollet фактор стабильности.  $\Delta$  означает определитель **S**.

Точки окружности могут быть заданы углом вектора **v**, где угол должен быть в градусах. Другая возможность – это специфицировать число **n** угловых равноудаленных точек окружности. Если не задан дополнительный аргумент **X**, берется 64 точки. Допустимое усиление мощности может быть также специфицировано вектором **Gp**, приводя к генерации **m** окружностей, где **m** – это величина **Gp**.

Пожалуйста, обратитесь также к “Qucs -Technical Papers”, глава 1.5.

### Пример

```
v=GpCircle(S)
```

### См. также

```
GaCircle(), Rollet()
```

**Mu()**

**Mu** – это фактор стабильности двух-портовой матрицы S-параметров.

**Синтаксис**

```
y=Mu ( S )
```

**Аргументы**

Name	Type	Def. Range	Required
S	$\mathbb{R}^{2 \times 2 \times p}, \mathbb{C}^{2 \times 2 \times p}, \mathbb{R}^{2 \times 2}, \mathbb{C}^{2 \times 2}$	$]-\infty, +\infty[$	✓

**Описание**

Функция возвращает Mu фактор стабильности  $\mu$  усилителя, описанного двух-портовой матрицей S-параметров – S:

$$\mu = \frac{1 - |S_{11}|^2}{|S_{22} - S_{11}^* \Delta| + |S_{21} S_{12}|}$$

$\Delta$  означает определитель S.  
Усилитель безусловно стабилен, если  $\mu > 1$ .  
Для S как вектор или матрица уравнение выше применяется к подматрицам S.

**Пример**

```
m=Mu ( S )
```

**См. также**

```
Mu2(), Rollet(), StabCircleS(), StabCircleL()
```

**Mu2()**

**Mu'** фактор стабильности двух-портовой матрицы S-параметров.

**Синтаксис**

```
y=Mu2 ( S )
```

**Аргументы**

Name	Type	Def. Range	Required
S	$\mathbb{R}^{2 \times 2 \times p}, \mathbb{C}^{2 \times 2 \times p}, \mathbb{R}^{2 \times 2}, \mathbb{C}^{2 \times 2}$	$] - \infty, + \infty [$	<input checked="" type="checkbox"/>

**Описание**

Эта функция возвращает Mu’ фактор стабильности  $\mu'$  усилителя, описанного двух-портовой матрицей S-параметров – S:

$$\mu' = \frac{1 - |S_{22}|^2}{|S_{11} - S_{22}^* \Delta| + |S_{21} S_{12}|}$$

$\Delta$  означает определитель S.

Усилитель безусловно стабилен, если  $\mu' > 1$ .

Для S как вектор или матрица уравнение выше применяется к подматрицам S.

**Пример**

```
m=Mu2 ( S )
```

**См. также**

```
Mu2(), Rollet(), StabCircleS(), StabCircleL()
```

## NoiseCircle()

Генерирует окружность(и) с постоянным шумовым образом(и).

### Синтаксис

```
y=NoiseCircle(Sopt,Fmin,Rn,F,v)
y=NoiseCircle(Sopt,Fmin,Rn,F,n)
y=NoiseCircle(Sopt,Fmin,Rn,F)
```

### Аргументы

Name	Type	Def. Range	Required	Default
Sopt	$\mathbb{R}^n, \mathbb{C}^n$	$]-\infty, +\infty[$	✓	
Fmin	$\mathbb{R}^n$	$[1, +\infty[$	✓	
Rn	$\mathbb{R}^n, \mathbb{C}^n$	$[0, +\infty[$	✓	
F	$\mathbb{R}, \mathbb{R}^n$	$[1, +\infty[$	✓	
v	$\mathbb{R}^n$	$[0, 360]^o$		
n	$\mathbb{N}$	$[2, +\infty[$		64

### Описание

Функция генерирует точки окружности постоянного образа шумов (Noise Figure – NF) F в комплексной плоскости источника (rS) усилителя. Обычно усилитель имеет минимум NF  $F_{min}$ , если коэффициент отражения источника  $rS = S_{opt}$  (шумовое согласование). Заметьте, что это состояние с оптимальным коэффициентом отражения источника  $S_{opt}$  отличается от согласования мощности! Следовательно усиление мощности при согласовании шумов ниже, чем максимально достижимое усиление. Эти значения  $S_{opt}$ ,  $F_{min}$  и нормализованное эквивалентное шумовое сопротивление  $R_n/Z_0$  могут обычно браться из справочных данных усилителя.

Радиус r и центр c окружности постоянной NF вычисляются из следующего:

$$r = \frac{\sqrt{N^2 + N \cdot (1 - |S_{opt}|^2)}}{1 + N} \text{ and } c = \frac{S_{opt}}{1 + N}, \text{ with } N = \frac{F - F_{min}}{4 R_n} \cdot Z_0 \cdot |1 + S_{opt}|^2.$$

Точки окружности могут быть заданы углом вектора v, где угол должен быть в градусах. Другая возможность – это специфицировать число n угловых равноудаленных точек окружности. Если не задан дополнительный аргумент X, берется 64 точки.

Пожалуйста, обратитесь также к “Qucs -Technical Papers”, часть 2.2.

### Пример

```
v=NoiseCircle(Sopt,Fmin,Rn,F)
```

### См. также

GaCircle(), GpCircle()

# PlotVs()

Возвращает группу данных, базируемую на векторе или матрице векторов с зависимостью от заданного вектора.

## Синтаксис

```
y=PlotVs(X, v)
```

## Аргументы

Name	Type	Def. Range	Required
X	$\mathbb{R}^n, \mathbb{C}^n, \mathbb{R}^{m \times n \times p}, \mathbb{C}^{m \times n \times p}$	$]-\infty, +\infty[$	✓
v	$\mathbb{R}^n, \mathbb{C}^n$	$]-\infty, +\infty[$	✓

## Описание

Функция возвращает группу данных, базируемую на векторе или матрице векторов X с зависимостью от заданного вектора v.

## Пример

```
PlotVs(Gain,frequency/1E9) .
```

## См. также

## Rollet()

**Rollet** – фактор стабильности двух-портовой матрицы S-параметров.

### Синтаксис

`y=Rollet(S)`

### Аргументы

Name	Type	Def. Range	Required
S	$\mathbb{R}^{2 \times 2 \times p}, \mathbb{C}^{2 \times 2 \times p}, \mathbb{R}^{2 \times 2}, \mathbb{C}^{2 \times 2}$	$]-\infty, +\infty[$	✓

### Описание

Эта функция возвращает Rollet фактор стабильности K усилителя, описанного двух-портовой матрицей S-параметров – S:

$$K = \frac{1 - |S_{11}|^2 - |S_{22}|^2 + |\Delta|^2}{2 |S_{21}| |S_{12}|}$$

$\Delta$  означает определитель S.

Усилитель безусловно стабилен, если  $K > 1$  и  $|\Delta| < 1$ .

Заметьте, что большое K может ввести в заблуждение в случае много-каскадного усилителя, притворяясь необычайной стабильностью. Это конфликтует с действительностью, где усилитель с большим усилением обычно страдает от неустойчивости из-за паразитных связей.

Для S как вектор или матрицы уравнение выше применяется к подматрицам S.

### Пример

`K=Rollet(S)`

### См. также

`Mu()`, `Mu2()`, `StabCircleS()`, `StabCircleL()`

## StabCircleL()

**Окружность стабильности в плоскости нагрузки.**

### Синтаксис

```
y=StabCircleL(X)
y=StabCircleL(X,v)
y=StabCircleL(X,n)
```

### Аргументы

Name	Type	Def. Range	Required	Default
X	$\mathbb{R}^{2 \times 2 \times p}, \mathbb{C}^{2 \times 2 \times p}$	$] -\infty, +\infty[$	✓	
v	$\mathbb{R}^n$	$[0, 360]^\circ$		
n	N	$[2, +\infty[$		64

### Описание

Эта функция генерирует точки окружности стабильности на комплексной плоскости коэффициента отражения нагрузки (rL) усилителя. Усилитель описывается двух-портовой матрицей S-параметров – S.

Радиус r и центр c окружности вычисляются:

$$r = \left| \frac{S_{21} S_{12}}{|S_{22}|^2 - |\Delta|^2} \right| \text{ and } c = \frac{S_{22}^* - S_{11} \cdot \Delta^*}{|S_{22}|^2 - |\Delta|^2}$$

$\Delta$  означает определитель S.

Точки окружности могут быть заданы углом вектора v, где угол должен быть в градусах. Другая возможность – это специфицировать число n угловых равноудаленных точек окружности. Если не задан дополнительный аргумент X, берется 64 точки.

Если центр rL плоскости лежит внутри этой окружности и  $|S_{11}| \leq 1$ , тогда окружность стабильна для всех коэффициентов отражения внутри окружности. Если центр rL плоскости лежит вне окружности и  $|S_{11}| \leq 1$ , тогда окружность стабильна для всех коэффициентов отражения вне окружности (пожалуйста, посмотрите “Qucs -Technical Papers”, часть 1.5).

### Пример

```
v=StabCircleL(S)
```

### См. также

```
StabCircleS(), Rollet(), Mu(), Mu2()
```

## StabCircleS()

**Окружность стабильности в плоскости источника.**

### Синтаксис

```
y=StabCircleS(X)
y=StabCircleS(X,v)
y=StabCircleS(X,n)
```

### Аргументы

Name	Type	Def. Range	Required	Default
X	$\mathbb{R}^{2 \times 2 \times p}, \mathbb{C}^{2 \times 2 \times p}$	$] -\infty, +\infty[$	✓	
v	$\mathbb{R}^n$	$[0, 360]^\circ$		
n	N	$[2, +\infty[$		64

## Описание

Эта функция генерирует точки окружности стабильности в комплексной плоскости коэффициента отражения источника (rS) усилителя. Усилитель описывается двух-портовой матрицей S-параметров – S.

Радиус r и центр c этой окружности вычисляются следующим образом:

$$r = \left| \frac{S_{21} S_{12}}{|S_{11}|^2 - |\Delta|^2} \right| \text{ and } c = \frac{S_{11}^* - S_{22} \cdot \Delta^*}{|S_{11}|^2 - |\Delta|^2}$$

$\Delta$  означает определитель S.

Точки окружности могут быть заданы углом вектора v, где угол должен быть в градусах. Другая возможность – это специфицировать число n угловых равноудаленных точек окружности. Если не задан дополнительный аргумент X, берется 64 точки.

Если центр плоскости rS лежит внутри окружности и  $|S_{22}| \leq 1$ , тогда окружность представляет стабильность всех коэффициентов отражения внутри окружности. Если центр плоскости rS лежит вне окружности и  $|S_{22}| \leq 1$ , тогда окружность – стабильность для всех коэффициентов отражения вне окружности (посмотрите также “Qucs -Technical Papers”, часть 1.5).

## Пример

```
v=StabCircleS(S)
```

## См. также

```
StabCircleL(), Rollet(), Mu(), Mu2()
```