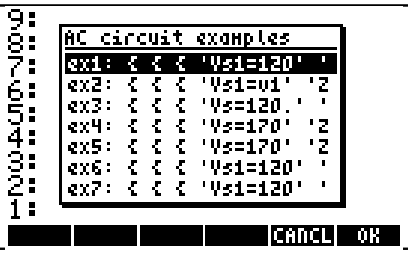
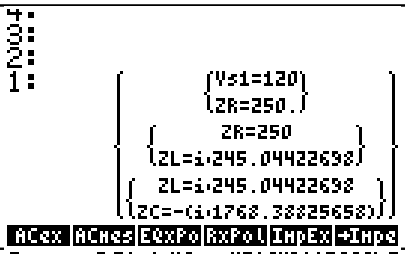
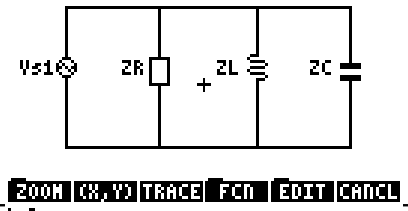
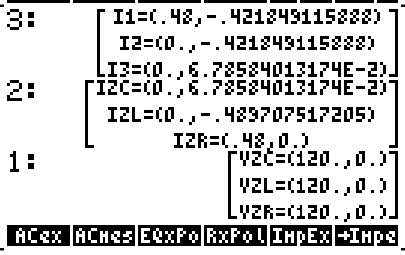
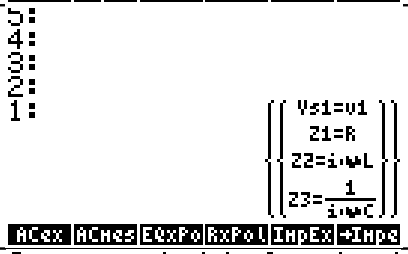
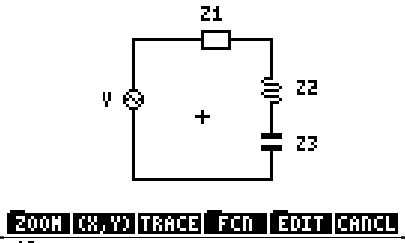
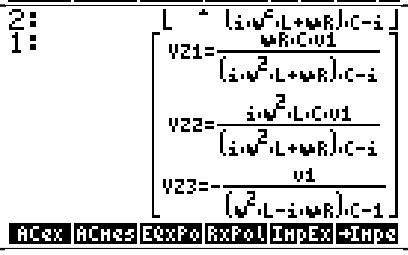
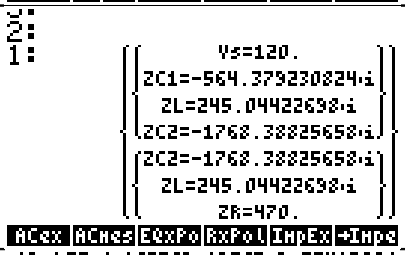
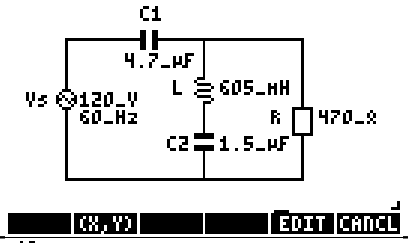
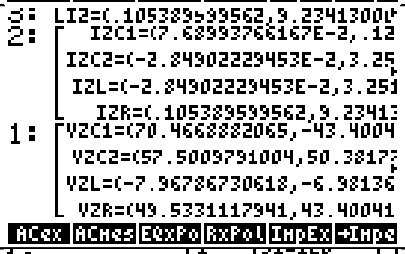
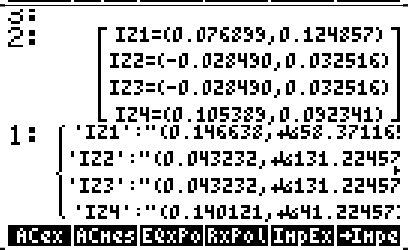
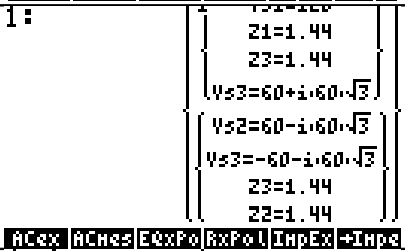
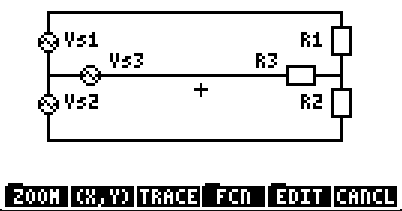
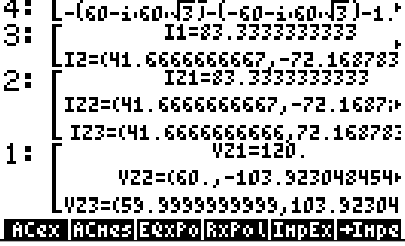
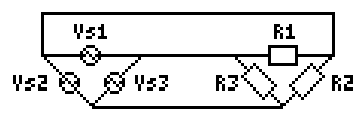
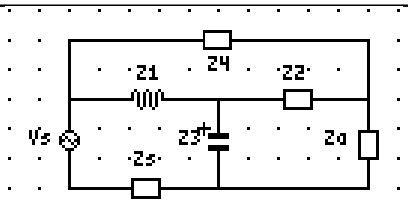
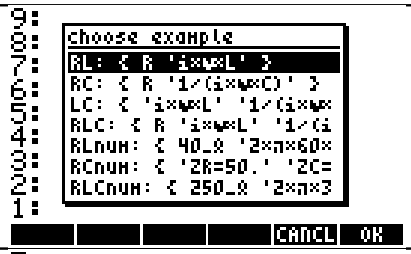
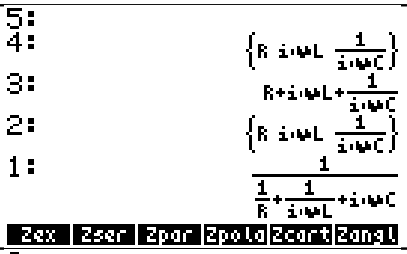
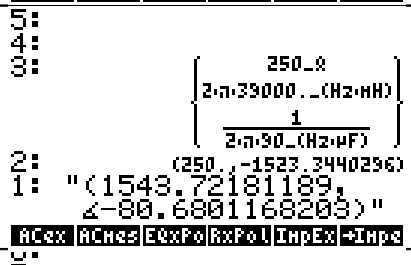
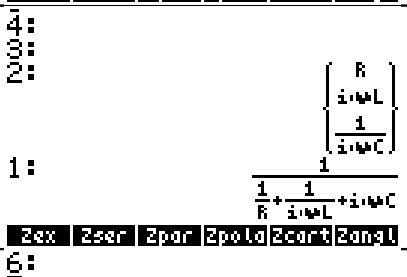
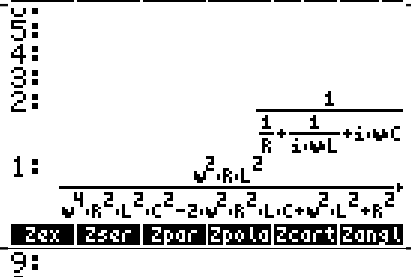
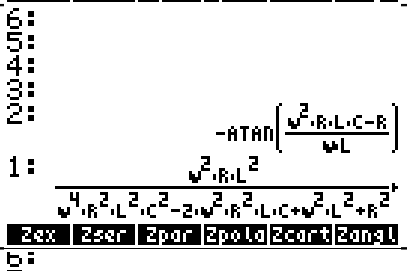
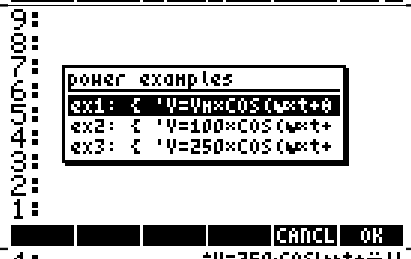
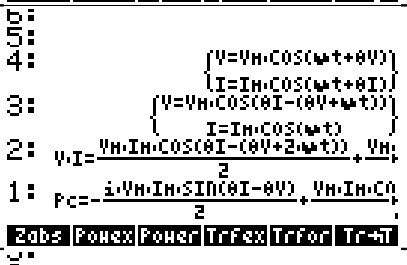
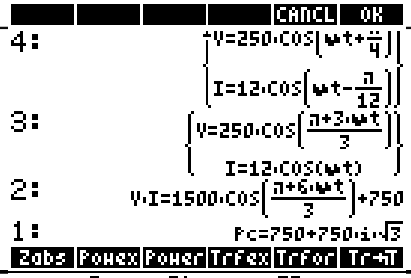
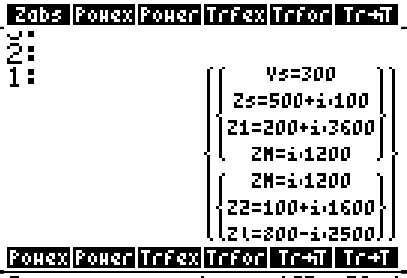
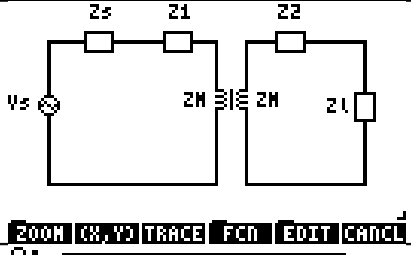
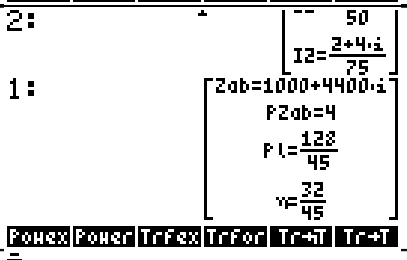
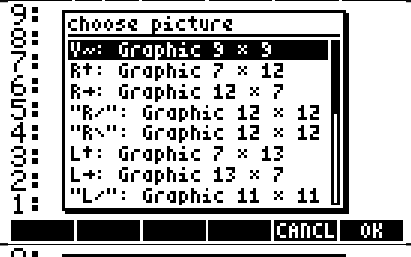
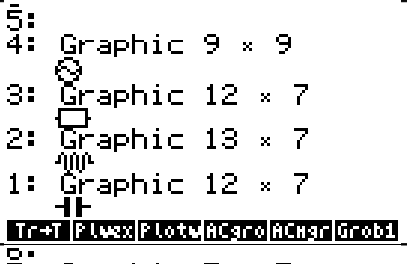
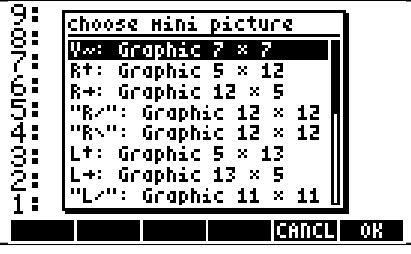
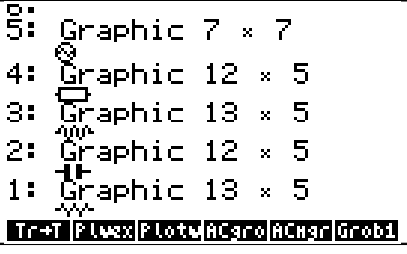
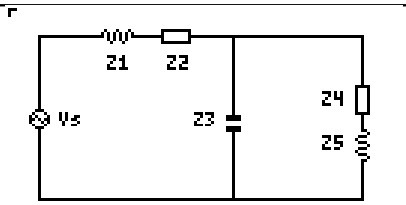
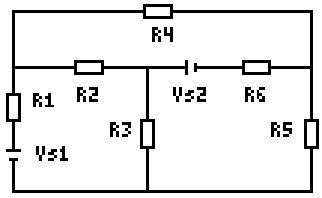
ACCIRC

ACex: examples for AC circuits [OK] shows list		
Cursor left shows the corresponding graph of circuit ACmesh: solves for currents and voltages in circuit (5s)		
an symbolic example graph of circuit		
ACmesh: solutions (10s) example 3		
graph of circuit ACmesh: solutions (4.3s)		
EQxPol: equation to polar form and back (0.6s) ACex: 3-phase example		
cursor left shows circuit ACmesh: calculate circuit (3.5s)		

Another 3-phase example graph	1: { { 'Vs1=120' 'Z1=1.44' } { 'Vs2=120×EXP(i×2×π/3)' 'Z2=1.44' } { 'Vs3=120×EXP(i×4×π/3)' 'Z3=1.44' } } +SKIP SKIP+ +DEL DEL+ DEL L INS	 (X,Y) EDIT CANCL
ACmesh solutions(5s) ACex: exmple	3: [I1=83.3333333333 I2=(-41.6666666663,72.16278 I3=(-41.6666666657,-72.16278 2: [I21=83.3333333333 I22=(-41.6666666663,72.1627 I23=(-41.6666666657,-72.1627 1: [VZ1=120. VZ2=(-59.9999999995,103.923 VZ3=(-59.9999999986,-103.923 ACex ACmes EQxPo RxPol ImpEx →Impe	1: { { 'Vs=170' 'Z1=i×6' 'Z3=-7×i' 'Zs=0' } { 'Z3=-7×i' 'Z2=10' 'Zg=0' } { 'Z4=24' 'Z2=10' 'Z1=i×6' } } +SKIP SKIP+ +DEL DEL+ DEL L INS
Graph of circuit ACmesh solutions(8s)	 ZOOM (X,Y) TRACE FCN EDIT CANCL	2: I2g=(13.467453505,-26.81336 I2s=(51.7721745351,-17.69313 1: VZ1=(106.158798283,268.1330 VZ2=(63.8412017167,-268.133(VZ3=(63.8412017166,-268.133(VZ4=170. VZg=(0.,0.) VZs=(0.,0.) ACex ACmes EQxPo RxPol ImpEx →Impe
ImpEx: examples for impedance calculation [OK] example with frequency ->Imped: calculates impedance (0.8s)	choose example ex1: { { 250_Ω 650_mH ex2: { { R L C } F } ex3: { { R1 L2 C3 C4 } ex4: { { 'R1=100_Ω' 'L ex5: { { 'R1=100_Ω' 'L ex6: { { { 'Vs=120_V' CANCL OK	3: { 250_Ω 650_mH 1.5_μF } 2: 60 1: { 250. i·245.04422698 -(i·1768.38825658) } ACex ACmes EQxPo RxPol ImpEx →Impe
symbolic examples	3: { R L C } 2: 'f' 1: { R i·2·π·f·L 1/(i·2·π·f·C) } ACex ACmes EQxPo RxPol ImpEx →Impe	3: { R1 L2 C3 } 2: 'ω' 1: { R1 i·ω·L2 1/(i·ω·C3) } ACex ACmes EQxPo RxPol ImpEx →Impe
example with ω ->Imped: calculate impedances (2s)	2: { R1=100_Ω L=10_mH R2=50_Ω C=10_μF } 1: ω=10000._1/s ACex ACmes EQxPo RxPol ImpEx →Impe	1: { ZR1=100. ZL=100.·i ZR2=50. ZC=-10.·i } ACex ACmes EQxPo RxPol ImpEx →Impe
example with f ->Imped: calculate impedances (2s)	2: { R1=100_Ω L=10_mH R2=50_Ω C=10_μF } 1: f=1591.54943092 ACex ACmes EQxPo RxPol ImpEx →Impe	1: { ZR1=100. ZL=100.·i ZR2=50. ZC=-10.·i } ACex ACmes EQxPo RxPol ImpEx →Impe
example with circuit (see example 3 above) ->Imped: calculate impedances (3.7s)	2: { { Vs=120_V C1=4.7_μF L=650_mH C2=1.5_μF } { C2=1.5_μF L=650_mH R=.47_kΩ } } 1: f=60_Hz ACex ACmes EQxPo RxPol ImpEx →Impe	1: { { Vs=120. ZC1=-564.379230824·i ZL=245.04422698·i ZC2=-1768.38825658·i } { ZC2=-1768.38825658·i ZL=245.04422698·i ZR=470. } } ACex ACmes EQxPo RxPol ImpEx →Impe

Zex: impedance examples RxPol: polar form (0.1s) Zser,Zpar: serial, parallel impedance (0.1s)		
Zser: serial impedances (1s) RxPol: polar form (0.1s) Zpar: parallel impedance (0.1s)		
Zcart: cartesian form (5s) Zangle: angle		
Powex: exampes for power calculation Power: power calculation (7s)		
Power: power calculation (7s) Trfex: example transformer		
graph of circuit Trformer: calulate circuit (5s)		
ACgrob: choose graphic object (grob) OK gives object		
ACmgrob: mini graphic objects OK gives object		

circuits drawn with minigrobs	 SUB REPL PICT→ R,Y→ PICT	 SUB REPL PICT→ R,Y→ PICT

ACinfo: info on circuits	Mesh current (loop) analysis
	for planar AC circuits:
	
	Input of AC-circuit:
	{
	{ 'Vs1=v1' 'ZM=ZM'.. }
	;
	{ 'Vsk=vk' 'Zl=r\'.. }
	}
	
	The complex impedances must be
	named as Z1,Z2.. ZR,ZL,ZC..
	GRAPH OK
		ex: ZR=5, ZL=i*5, ZC=-i*6
		
		We recommend to start
		from the lower left corner and
		to proceed clockwise for every
		mesh. Then it should be easy to
		draw the circuit from the list
		of lists.
		Loop currents are named I1,I2..
		in the order of the lists,
		Complex impedances are Z1,Z2..,
		voltage sources are Vs1,Vs2..
		GRAPH OK

HelpAC: help	AC alternate current ACex _ → {}..{}} choosebox With examples for ACMesh. press cursor left to see a graphic of the circuit ACMesh {{}..{}} → {} [Ikeq] [Ik=..] [IRk=..] [VRk=..] mesh analysis for planar AC networks. In the lists use impedances ZR=R, ZL=i*w*L, ZC=-i/(w*C) GRAPH OK	w=2*π*f, f=frequency details see ACinfo EQxPol ['I1=(1,2)'..] ↔ {:I1:(2.23,∠63.4)} convert num equation with complex nr to polar form RxPol {(x,y)..} ↔ {(r,∠)} complex nrs to polar ImpEx _ → {} example for →Imped →Imped {} f,w → {} impedance from R, L, C and frequency f,w You can enter the circuit GRAPH OK
AChelp:	with Vs,R,L,C and →Imped calculates the list with impedances for ACMesh ex 6 in Impex gives ex 3 in ACex Zex _ → {} examples resistors Zser {} → Σ serial resistors Zpar {} → Σ parallel resistors Zpolar x+iy → r*exp(i*∠) Zcart (x,y) → x+iy Zangle (x,y) → ∠ (x,y) Zabs (x,y) → √(x^2+y^2) (x,y) GRAPH OK	Powex _ → {} example for Power Power {} → V*I=.., Pc calculates time dependent and complex power from sin, cos voltage, current [LS] [EXPtLN] TEXPAND gives the SIN,COS part Pc=P+i*Q P=average, Q=reactive power P,Q=V0*I0/2*COS,SIN(∠v-∠i) Trfex _ → {} example TrFormer press cursor left for GRAPH OK
AChelp:	picture, ZM=i*w*M, M=k*√(L1*L2), 0<k≤1 TrFormer {} → {} [eqn] [Ik] [Zab PZab Pl η] calculate lin.transformer Ik=current, Zab=impedance at source, P=RE(Z)*|I|^2 power at Zab,Zl=load, η=Pl/PZab Tr→T {L1 L2 M} → {La Lb Lc} Pi EQUIVALENT CIRCUIT Tr→T {L1 L2 M} → {La Lb Lc} GRAPH OK	T EQUIVALENT CIRCUIT Plwex _ → examples for Plotw Plotw a(w)+i*b(w),a(f)+i*b(f) → parametric plot AC(M)Grob _ → grob, choose (mini)grob to plot circuit Grob1 var,string → grob size 1 Parreset _ → _ reset to parameter plot Cflags _ → _ reset to flags for easy handling of GRAPH OK