

amort-prepay-v06

Rei Shinozuka rei@reishinozuka.com

2025-04-26

Amortization and Prepayment Calculations for Fixed-Rate Level-Pay Mortgages

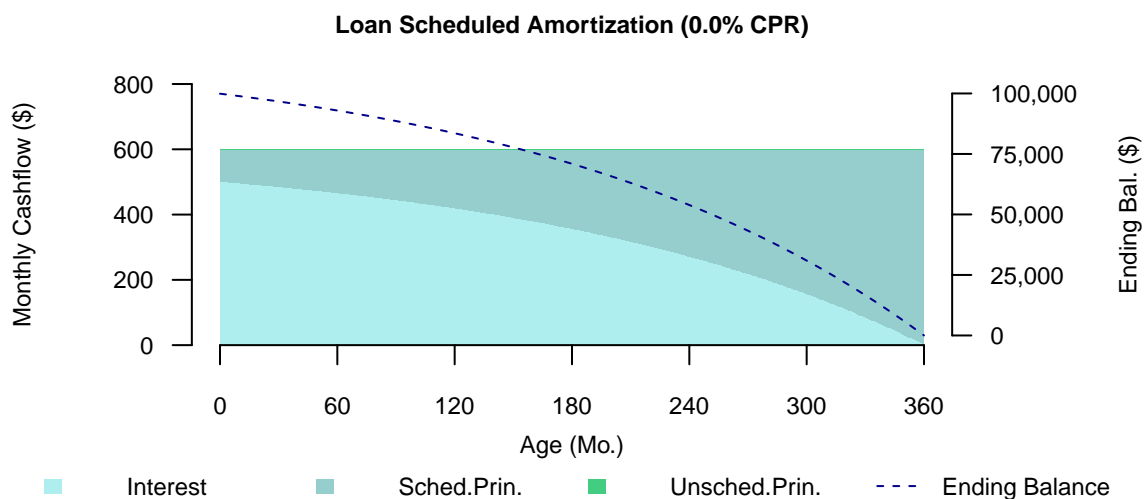
Description and Definitions

Fixed-Rate Level-Pay mortgages comprise the great majority of loans originated in the US.

- Fixed-Rate: constant interest rate over the life of the loan
- Level-Pay: Loan is structured such that monthly payment is constant over the life of the loan.

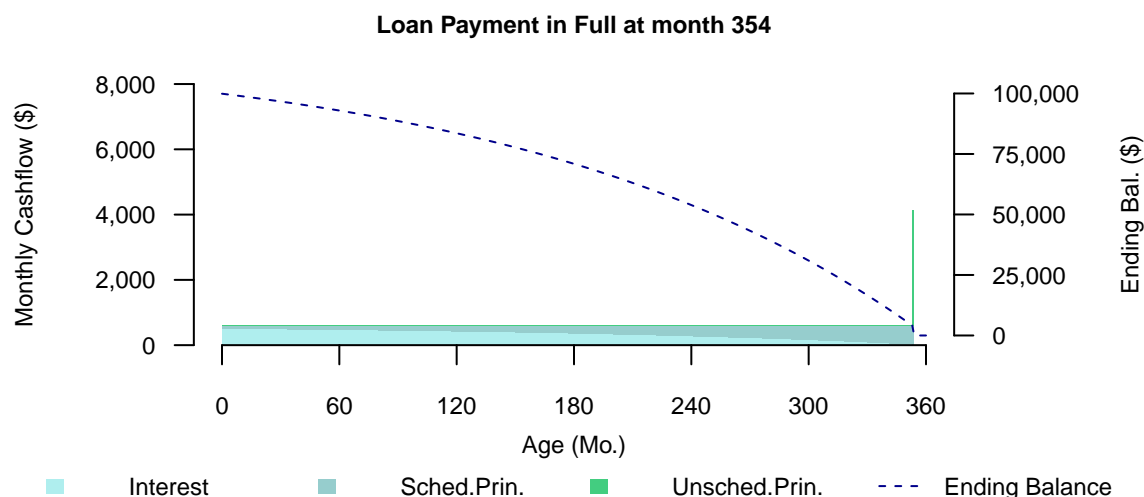
The level-pay structure is facilitated by *amortizing* the balance over the life of the loan after paying monthly interest. The amount of the amortization is known as the *scheduled principal* and the remaining balance is known as the *scheduled balance*.

The below example shows scheduled amortization. Principal and interest equal the scheduled payment with interest comprising the majority of the monthly payment in the early part of the loan. The payment window is the term of the loan.



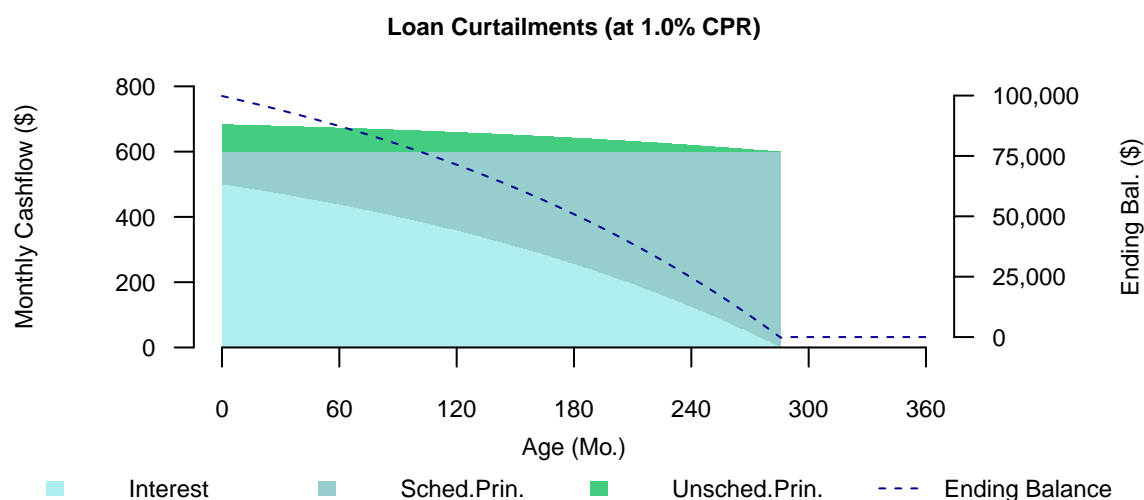
The borrower has the right to pay off the full balance of the loan at any time, to facilitate a refinancing or property sale, resulting in the loan being *paid in full*.

The below examples shows a loan paying in full prior to loan maturity.



The borrower may also partially prepay the remaining balance of the loan resulting in a *curtailment*. Both payments-in-full and curtailments are types of *prepayments*, also known as *unscheduled principal* payments.

The below example shows loan which curtails at a constant rate. The scheduled payment (interest plus scheduled principal) never changes. The curtailments reduce the duration of the payment window.



The growth of Mortgage-Based Securities (MBS) in the 1980s led to investors to evaluate mortgage loans as pools of loans. The borrower's option to refinance the loan is to the investor a short call option. Reflecting this exposure, MBS investor reporting attempts to describe historical prepayments and project future prepayments. The key measure of pool prepayments is *single monthly mortality (SMM)* which is the monthly ratio of aggregate unscheduled principal payments to the pool's scheduled ending balance. More commonly, prepayments are reported as a *conditional prepayment rate (CPR)* which is the compounded annualized SMM.

The CPR assumption used to project pool cashflows abstracts the mortgage pool into a homogeneous asset which any proportion of remaining balance may be prepaid. This abstraction does not recognize individual loans within the pool. Rather, it assumes the pools consist of a very large number of infinitesimally sized loans. Actual behavior is that individual borrowers decide on a monthly basis to prepay their loans. The convention adopted by MBS investors assumes continuous prepayments to estimate the discrete behavior of actual loan pools. The difference between the continuous and discrete view is minimized when pools consist of the large number of loans (>250 loan count) typically required for MBS securitization.

To illustrate the distinction, this report shows:

- A. Single Loan Amortization and Prepayment
- B. Pool Amortization and Prepayment under MBS Convention
- C. Pool Amortization and Prepayment simulating discrete loans

Parameters

- Original Term = 360
- Note rate = 6.00%
- Original Loan Balance = 100,000
- Monthly payment = 599.55
- Conditional Prepayment Rate (CPR) = 6.00%
- Pool Loan Count = 50

A. Single Loan Amortization and Prepayment

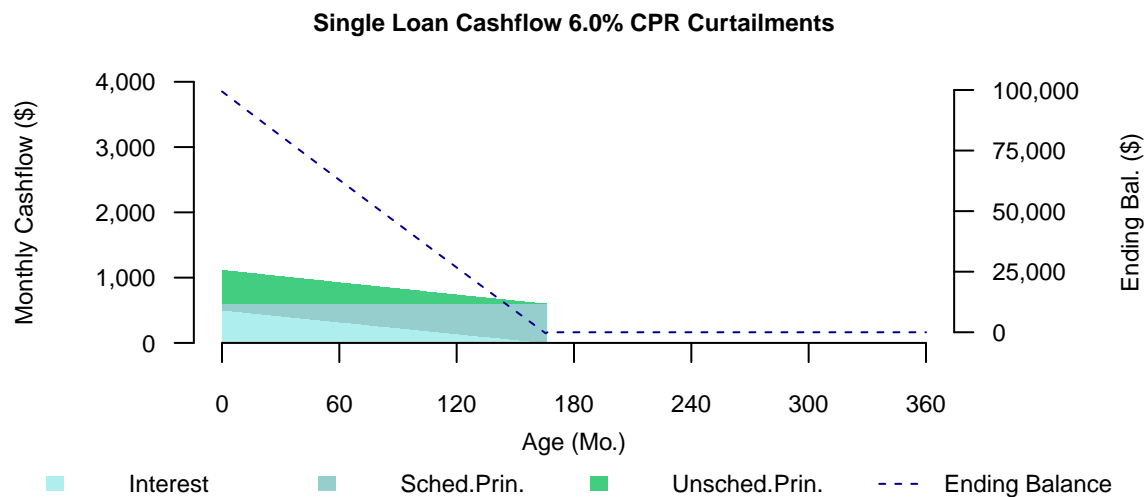
The single loan is modeled following its contractual terms.

CPRs for single loans are treated as:

1. Scheduled amortization (CPR == 0%)
2. Curtailments ($0\% < \text{CPR} < 100\%$) or
3. Payments in full (CPR == 100%)

Other characteristics of single loan amortization:

1. the Scheduled Payment does not change
2. Coupon does not change over time
3. Age and Remaining Term increment by 1 and -1 respectively every month
4. Nonzero $\text{CPR} < 100\%$ represents curtailments, which
 - a. reduce the payment window by accelerating the final paydown to zero
 - b. do not affect scheduled (level) payment



B. Pool Amortization and Prepayment under MBS Convention

The MBS convention assumes prepayments are *continuous*, implying pools consist of infinite number of loans each of *infinitesimal size*. There is no explicit modeling of loans and loan counts, though the latter may be inferred if average loan size is an attribute of the pool.

Balances and other quantities represent the aggregated values of pseudo-loans comprising the pool.

CPRs under MBS pool convention is treated as:

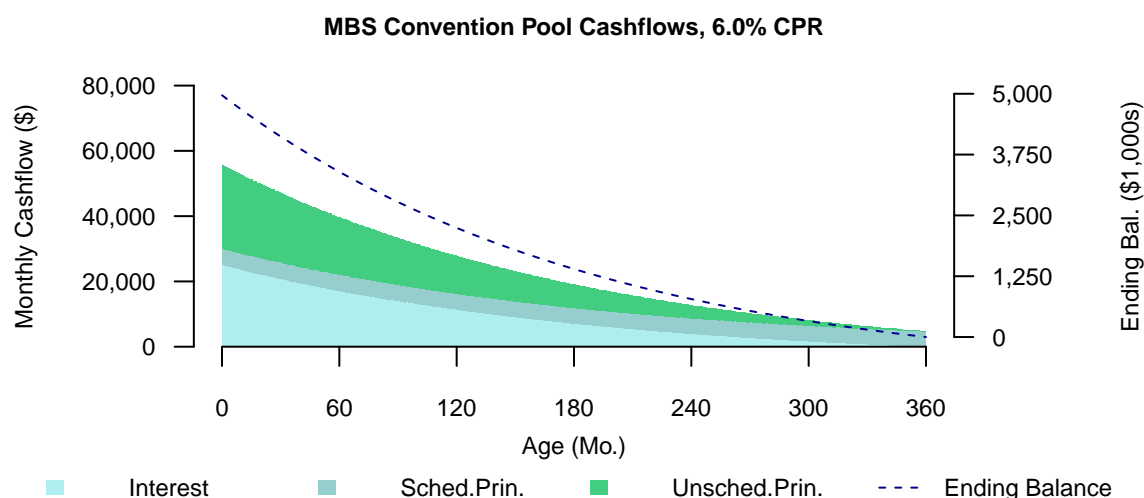
1. paid in full for a proportion of loans up to 100% (CPR > 0%)
2. scheduled amortization (CPR == 0%)
3. There is no provision for curtailment.

Example:

1. Assume flat 6.00% CPR
2. 6.00% CPR is annualized equivalent of 0.51% Single Monthly Mortality (SMM).
3. Each month, 0.51% of the loans are assumed to pay in full, reducing actual balances by that amount
4. The scheduled level payment is also reduced by 0.51% reflecting the paydowns.

Other characteristics of the pool:

1. the scheduled payment changes
2. coupon does not change over time
3. age and remaining term increment by 1 and -1 respectively every month
 - a. age represents Weighted Average Loan Age (WALA)
 - b. remaining term represents Weighted Average Maturity (WAM)
4. Prepayments do not affect the payment window



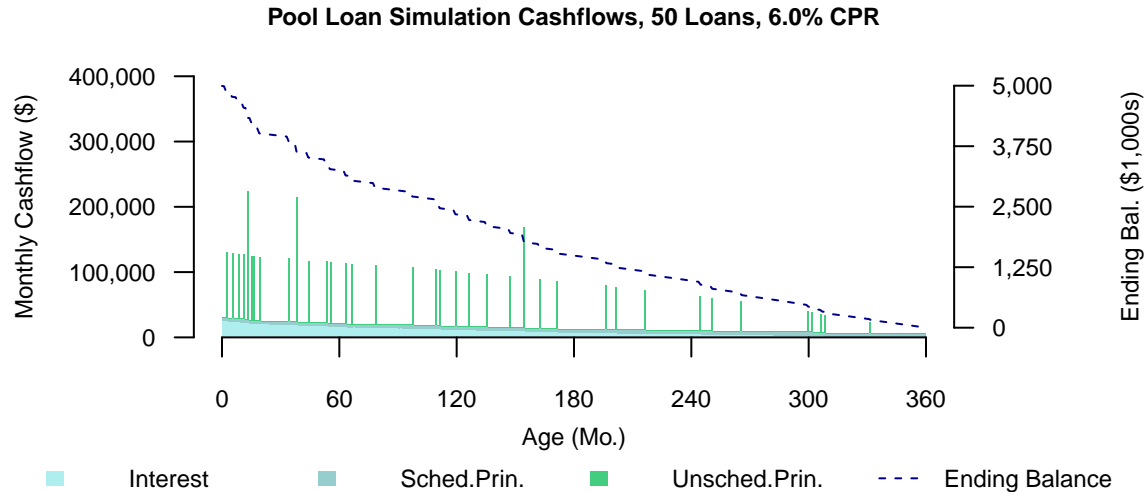
C. Pool Amortization and Prepayment simulating discrete loans

The simulation models a pool as an aggregate of a *specific number* of loans of *specific size*. Though not captured in this simulation, loans may have heterogeneous properties (different coupons, sizes, etc).

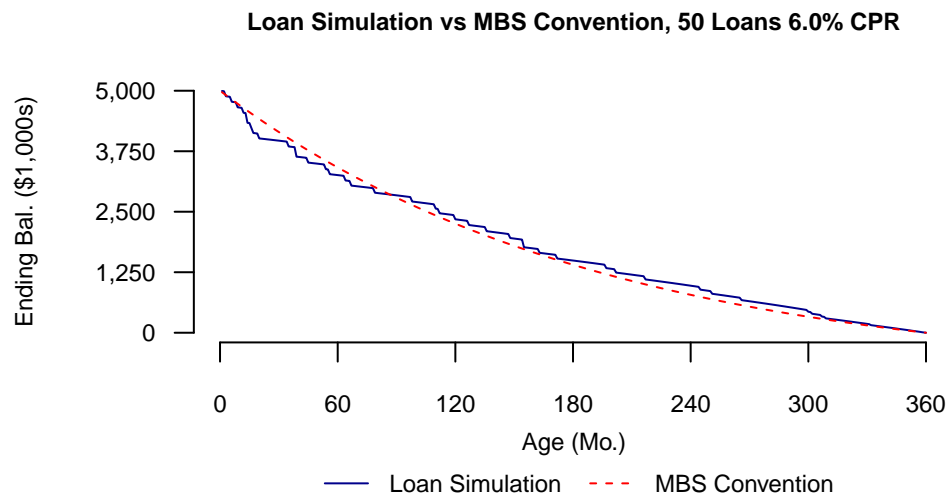
The simulation interprets an SMM as the probability for an individual loan being paid in full each month. There is no provision for curtailment.

Example:

1. Assume flat 6.00% CPR
2. 6.00% CPR is annualized equivalent of 0.51% Single Monthly Mortality (SMM).
3. Each month, each loan has a 0.51% probability of paying in full.
4. The scheduled level payment is reduced by the scheduled level payments of those loans paying in full.



Comparing Simulation against MBS Convention With higher loan counts, the discrete simulation approaches results of the MBS convention.



Amortization Functions

```
# level.pay.calc()
# returns level (scheduled) payment given term to maturity and note.rate
level.pay.calc<-function(rem.term,monthly.rate)
{
  monthly.payment<-(monthly.rate*(1+monthly.rate)^rem.term)/((1+monthly.rate)^rem.term-1)
```

```

    return(monthly.payment)
}# level.pay.calc()

# amortize single loan incorporating prepayments
# CPR == 1 signifies paid-in-full
# 0 < CPR < 1 are curtailments
amort.loan.cf<-function(orig.term,note.rate,orig.bal,cpr=c(NA))
{
  monthly.rate<-note.rate/12
  if(is.na(cpr[1])){
    smm<-replicate(orig.term,0)
  } else {
    smm<-1-(1-cpr)^(1/12)
  }
  level.pay<-level.pay.calc(orig.term,monthly.rate)*orig.bal
  # initialize data.table for output
  cf<-data.table(mon=1:orig.term)
  cf[1,bbal:=orig.bal]
  cf[,int:=0]
  cf[,sched.prin:=0]
  cf[,unsched.prin:=0]
  cf[,prin:=0]
  cf[,eschedbal:=0]
  cf[,eбал:=0]
  # begin amortization
  mo.bbal<-orig.bal
  mo.unsched.prin<-0
  for(m in 1:orig.term){
    mo.int<-mo.bbal * monthly.rate           # pay interest first
    mo.sched.prin<-level.pay - mo.int        # remainder of payment is sched prin
    mo.eschedbal<-mo.bbal - mo.sched.prin    # balance after sched prin is sched bal
    mo.unsched.prin<-mo.eschedbal*smm[m]     # unsched prin is % of sched bal
    mo.prin<-mo.sched.prin + mo.unsched.prin # total prin is sched + unsched prin
    mo.eбал<-mo.bbal - mo.prin               # ending actual balance
    # copy results to data.matrix, rounding to $0.01
    cf[m,bbal:=round(mo.bbal,2)]
    cf[m,int:=round(mo.int,2)]
    cf[m,sched.prin:=round(mo.sched.prin,2)]
    cf[m,unsched.prin:=round(mo.unsched.prin,2)]
    cf[m,prin:=round(mo.prin,2)]
    cf[m,eschedbal:=round(mo.eschedbal,2)]
    cf[m,eбал:=round(mo.eбал,2)]
    if(mo.eбал<=0.005) break
    mo.bbal<-mo.eбал # for next month
  }
  return (cf)
} # amort.loan.cf()

```

Single Loan Amortization

```

# amort.pool.cf()
# Amortize loan pool using MBS convention

```

```

# which models prepayments as continuous
# and pool as consisting of many loans of 0.01 each
# CPR in a given month represents the % of loans that PAY IN FULL that month
# there is no provision for curtailments
amort.pool.cf<-function(orig.term,note.rate,orig.bal,cpr=c(NA))
{
  monthly.rate<-note.rate/12
  if(is.na(cpr[1])){
    smm<-replicate(orig.term,0)
  } else {
    smm<-1-(1-cpr)^(1/12)
  }
  # initialize data.table for output
  cf<-data.table(mon=1:orig.term)
  cf[,bbal:=orig.bal]
  cf[,int:=0]
  cf[,sched.prin:=0]
  cf[,unsched.prin:=0]
  cf[,prin:=0]
  cf[,eschedbal:=0]
  cf[,ebal:=0]
  mo.bbal<-orig.bal
  mo.unsched.prin<-0
  for(m in 1:orig.term){
    level.pay<-level.pay.calc(orig.term+1-m,monthly.rate)*mo.bbal # recalc level pay based on act bal
    mo.int<-mo.bbal * monthly.rate # pay interest first
    mo.sched.prin<-level.pay - mo.int # remainder is sched prin
    mo.eschedbal<-mo.bbal - mo.sched.prin # bal after sched prin is sched bal
    mo.unsched.prin<-mo.eschedbal*smm[m] # unsched prin is % of sched bal
    mo.prin<-mo.sched.prin + mo.unsched.prin # total prin = sched + unsched prin
    mo.ebal<-mo.bbal - mo.prin # actual bal
    # copy results to data.matrix, rounding to $0.01
    cf[,bbal:=round(mo.bbal,2)]
    cf[,int:=round(mo.int,2)]
    cf[,sched.prin:=round(mo.sched.prin,2)]
    cf[,unsched.prin:=round(mo.unsched.prin,2)]
    cf[,prin:=round(mo.prin,2)]
    cf[,eschedbal:=round(mo.eschedbal,2)]
    cf[,ebal:=round(mo.ebal,2)]
    if(mo.ebal<0.005) break # half-cent tolerance
    mo.bbal<-mo.ebal # for next month
  }
  return (cf)
} # amort.pool.cf

```

MBS Convention Pool Amortization

Parameters

```

orig.term<-360
note.rate<-0.06
orig.bal<-100000
scalar.cpr<-0.06
cpr<-replicate(orig.term,scalar.cpr)

```

```

# parameter string equivalents
scalar.smm<-1-(1-scalar.cpr)^(1/12)
note.rate.str<-sprintf("%5.2f%%",note.rate*100)
orig.bal.str<-prettyNum(orig.bal,big.mark="," scientific=FALSE)

scalar.cpr.str<-sprintf("%5.2f%%",scalar.cpr*100)
scalar.smm.str<-sprintf("%5.2f%%",scalar.smm*100)

loan.count<-50

```

Run Cashflows

```

cpr.eg<-0
cprsingle<-replicate(orig.term,cpr.eg)
cfloan<-amort.loan.cf(orig.term,note.rate,orig.bal,cprsingle)
title<-sprintf("Loan Scheduled Amortization (%3.1f%% CPR)",cpr.eg*100)
plot.cfs(cfloan,title)

pif.mon<-354
cprsingle<-replicate(orig.term,0)
cprsingle[pif.mon]<-1.0
cfloan<-amort.loan.cf(orig.term,note.rate,orig.bal,cprsingle)
title<-sprintf("Loan Payment in Full at month %3d",pif.mon)
plot.cfs(cfloan,title)

cpr.eg<-0.01
cprsingle<-replicate(orig.term,cpr.eg)
cfloan<-amort.loan.cf(orig.term,note.rate,orig.bal,cprsingle)
title<-sprintf("Loan Curtailments (at %3.1f%% CPR)",cpr.eg*100)
plot.cfs(cfloan,title)

cf1<-amort.loan.cf(orig.term,note.rate,orig.bal,cpr)
title<-sprintf("Single Loan Cashflow %3.1f%% CPR Curtailments",scalar.cpr*100)
plot.cfs(cf1,title)

pool.orig.bal<-orig.bal*loan.count
cf2<-amort.pool.cf(orig.term,note.rate,pool.orig.bal,cpr)

title<-sprintf("MBS Convention Pool Cashflows, %3.1f%% CPR",scalar.cpr*100)
plot.cfs(cf2,title)

```

```

set.seed(666)

cpr<-replicate(orig.term,scalar.cpr)
smm<-1-(1-cpr)^(1/12)

for(loan.no in 1:loan.count){
  pifcpr<-runif(n=orig.term) # uniformly distributed random
  pifcpr<-(pifcpr<smm)*1     # interpret smm as a probability
  cf1<-amort.loan.cf(orig.term,note.rate,orig.bal,pifcpr)
  # accumulate individual loan cashflows to the pool level
  if(loan.no==1){
    accum_cf<-cf1
  }
}

```



```

} else {
  accum_cf<-accum_cf+cf1
}
}
accum_cf$mon<-accum_cf$mon/loan.count

```

```

title<-sprintf("Pool Loan Simulation Cashflows, %d Loans, %3.1f%% CPR",loan.count,scalar.cpr*100)
plot.cfs(accum_cf,title)

```

Loan Simulation Pool Amortization

Plot Functions

```

# plot.cfs() plot cashflows
#
plot.cfs<-function(cf,title){
  par(mar=c(bottom=5, left=5, top=3, right=5) + 0.1) # bot, left, top, right in lines
  maxy<-scale.axis.max(max(cf$int+cf$sched.prin+cf$unsched.prin))
  displ.y<-maxy
  suff.y<-"$"
  if(maxy>1000*1000){
    displ.y<-maxy/1000
    suff.y<-"$1,000s"
  }
  cfm<-as.matrix(cf[,c("int", "sched.prin", "unsched.prin")])
  rownames(cfm)<-cf$mon
  cfmt<-t(cfm) # transpose
  barplot(cfmt,border=NA,col=c(int.col,sched.prin.col,unsched.prin.col),space=c(0,0),
    ylim=c(0,maxy),
    cex.main=cex.size,
    xaxt="n", yaxt="n", xlab="", ylab="", main=title)
  axis(1, at = seq(0, orig.term, 12*5),cex.axis=cex.size)
  axis(2, at = seq(0, maxy, maxy/4),
    prettyNum(seq(0, displ.y, displ.y/4),big.mark=",", scientific=FALSE),las=1,cex.axis=cex.size)

  maxy2<-max(cf[!is.na(cf$bbal)]$bbal)

  displ.y2<-maxy2
  suff.y2<-"$"
  if(maxy2>1000*1000){
    displ.y2<-maxy2/1000
    suff.y2<-"$1,000s"
  }

  par(new=TRUE) # new plot on same canvas, for ending balance
  plot(cf$mon,cf$ebal,type="l",col=ebal.col,xlab="", lty=2, ylab="",xaxt="n",yaxt="n",axes=FALSE,
    cex.main=cex.size ## cex.axis=cex.size,cex.lab=cex.size
  )
  axis(4, at = seq(0, maxy2, maxy2/4),
    prettyNum(seq(0, displ.y2, displ.y2/4),big.mark=",", scientific=FALSE),las=1,cex.axis=cex.size)

  mtext("Age (Mo.)",side=1,line=2.0,cex=cex.size)
  mtext(paste("Monthly Cashflow",suff.y),side=2,line=4.0,cex=cex.size)

```

```

mtext(paste("Ending Bal.", suff.y2), side=4, line=4.0, cex=cex.size)
legend(x="bottom", xpd=TRUE, inset = c(0, -0.65), horiz=TRUE, cex=cex.size,
      legend=c("Interest", "Sched.Prin.", "Unsched.Prin.", "Ending Balance"),
      lty=c(1,1,1,2),
      col=c(NA,NA,NA,ebal.col),
      fill=c(int.col,sched.prin.col,unsched.prin.col,NA), border=NA,
      bty="n"
    )
} # plot.cfs()

plot.comp.bals<-function(cf,cf2,title){
  par(mar=c(bottom=5, left=5, top=3, right=5) + 0.1) # bot, left, top, right in lines

  maxy2<-max(cf[!is.na(cf$bbal)]$bbal)

  maxy<-max(cf[!is.na(cf$bbal)]$bbal, cf2[!is.na(cf2$bbal)]$bbal)
  displ.y<-maxy
  suff.y<-"($)"
  if(maxy>1000*1000){
    displ.y<-maxy/1000
    suff.y<-"$1,000s"
  }

  plot(cf$mon, cf$ebal, type="l", col=ebal.col, main=title,
       xlab="", ylab="", xaxt="n", yaxt="n", axes=FALSE,
       cex.main=cex.size)
  lines(cf2$mon, cf2$ebal, lty=2, col=ebal2.col) # dashed
  axis(1, at = seq(0, orig.term, 12*5), cex.axis=cex.size)
  axis(2, at = seq(0, maxy, maxy/4),
       prettyNum(seq(0, displ.y, displ.y/4), big.mark=",", scientific=FALSE), las=1, cex.axis=cex.size)

  mtext("Age (Mo.)", side=1, line=2.0, cex=cex.size)
  mtext(paste("Ending Bal.", suff.y), side=2, line=4.0, cex=cex.size)
  legend(x="bottom", xpd=TRUE, inset = c(0, -0.65), horiz=TRUE, cex=cex.size,
       legend=c("Loan Simulation", "MBS Convention"),
       lty=c(1,2),
       col=c(ebal.col, ebal2.col),
       bty="n"
    )
} # plot.comp.bals()

```