Session _____

THE SAVE-SPEND PROBLEM

Mohammad H. Alimi, and Howard B. Wilson
North Dakota State University / University of Alabama

Introduction

Applied mathematics courses usually include topics from physics and engineering but seldom consider financial concepts. This omission is easy to remedy because the essential ideas of investment growth resulting from compounded investment earnings can be explained with a simple first-order differential equation. The current article analyzes what can be appropriately called *the save-spend problem* where funds earning interest are saved over one period and are consumed during a subsequent period. Special cases of this problem include mortgage financing as well as pension saving.

Mathematical Formulation

Investment capital Q growing due to a saving rate $S(t)$ while simultaneously earning a continuously compounded, after-tax, rate of investment return R satisfies the differential equation

$$Q'(t) = RQ(t) + S(t), \qquad Q(0) = q_0 .$$

The general solution of this equation for constant R is

$$Q(t) = e^{Rt}[q_0 + \int_0^t e^{-Rt} S(t)dt]$$

Inflation usually exists in real situations so it is desirable to think in terms of inflation adjusted capital defined by

$$q(t) = Q(t)e^{-It}$$

where I is the annual inflation rate. When I is zero, then q simply reduces to Q. The differential equation for $q(t)$ is

$$q'(t) = (R - I)q + S(t)e^{-It}, \quad q(0) = q_0$$

The form used here for $S(t)$ models a saving period followed by a payout period so that

$$S(t) = se^{At}, \quad 0 \le t \le t_1 \text{ and } S(t) = -pe^{At}, \quad t > t_1$$

where s, p and A are constants. Constants s and p are called the saving and payout rates. Parameter A, referred to as the saving growth constant, quantifies the rate at which $S(t)$ changes to account for inflation and wage growth. Common choices for A include $A = 0$ and $A = I$. The differential equation for $q(t)$ becomes

$$q'(t) = rq(t) + se^{at} - (s + p)e^{at}(t > t_1)$$

where $r = R - I$, $a = A - I$, and $(t > t_1)$ means one for $t > t_1$ and zero otherwise. Integrating the differential equation gives

$$q(t) = q_0 e^{rt} + \frac{s}{u}[e^{rt} - e^{at}] - \frac{(s+p)}{u} e^{at_1}[e^{r(t-t_1)} - e^{a(t-t_1)}](t > t_1)$$

where $u = r - a$. The value of q at time t_1 is

$$q_1 = q_0 e^{rt_1} + \frac{s}{u}[e^{rt_1} - e^{at_1}]$$

This equation quantifies how savings grow through interest compounding. In a slightly different sense, it can be used to compute the payment needed to retire a mortgage loan. For example, reducing an initial debt amount of $-q_0$ to zero at time t_1 would require an annual payment of

$$ss = q_0 u / [1 - e^{-ut_1}]$$

Next, let us turn to the important case involving a saving phase for $0 \le t \le t_1$ and a payout phase for $t_1 < t \le T$ where $T = t_1 + t_2$. Denoting $q(t_1 + t_2)$ by q_2 gives

$$q_2 = q_0 e^{rT} + \frac{s}{u}[e^{rT} - e^{aT}] - \frac{(s+p)}{u} e^{at_1}[e^{rt_2} - e^{at_2}]$$

A slight rearrangement of the last equation leads to

$$s[e^{ut_1} - 1] - p[1 - e^{-ut_2}] = ue^{-rt_2 - at_1}[q_2 - q_0 e^{rT}]$$

where $u = r - a$. This equation, relating s, p, q_0 and q_2 linearly, can be solved for any one parameter when the other three are given. An important special case occurring when q_0 and q_2 are zero is

$$p = s\, g(u, t_1, t_2), \quad g(u, t_1, t_2) = [e^{ut_1} - 1]/[1 - e^{-ut_2}].$$

This equation determines the saving rate needed to produce a desired payout or the payout resulting for a given saving rate. The following three examples illustrate use of the formulas just developed.

1) What is the monthly payment needed to repay a $150,000 home mortgage financed at 8% over 30 years? Taking $q_0 = 0,\ R = 0.08,\ A = I = 0.0,\ t_1 = 30$ leads to

$$pp = \frac{(150000)(0.08)}{(12)(1 - e^{-(30)(0.08)})} = \$1099.77 \text{ per month}$$

This value is quite close to the monthly payment of $1100.66 obtained from mortgage tables.

2) Suppose someone wants to save a million dollars, inflation adjusted, over a 40 year period. Savings earn 8%, inflation is 4%, and the savings rate will be increased to match inflation. What monthly saving rate is needed? Taking

$$q_0 = \$1{,}000{,}000,\ R = 0.08,\ A = I = 0.04,\ r = 0.04,\ a = 0.00 \text{ leads to}$$

$$pp = \frac{(1E6)(0.04)}{(12)(e^{(0.04)(40)} - 1)} = \$843.23 \text{ per month.}$$

3) An engineer expects to work for 40 years and retire for 20 years. He wants to have an inflation protected pension of $2500 per month. Initial and final savings will be zero. If savings are expected to earn 8%, inflation is 4% and the saving rate will be increased to

match inflation, what monthly saving rate during the 40 year work period will suffice to support the desired pension? Taking $R = 0.08,\ \ A = I = 0.04,\ \ p = (2500)(12) = \$30000$ per year leads to $ss = \dfrac{(30000)(1-e^{-(0.04)(20)})}{(12)(e^{(.04)(40)}-1)} = \$348.25$ per month.

The above examples assume a conservative 8% rate of investment return. It may be worth noting that, over the past fifty years, the Dow Jones industrial average, with dividends reinvested, has earned an average annual return of about 12% while the rate of inflation has been close to 4%. Furthermore, a self employed worker earning \$40,000 per year now pays \$6120 annually in Social Security and Medicare taxes. Assume that worker will be employed for 40 years and will retire for 20 years. Someone investing \$6120 annually and achieving an investment return 4% above inflation, could fund an annual pension of \$43,900. Similarly, if the individual were fortunate enough to beat inflation by 6%, the pension would be \$87,700 per year! Presently, the maximum amount which one wage earner and his spouse can receive after age 65 is less than \$34,000 even though self employed workers earning as much as \$68,000 have to pay about \$10,000 per year in Social Security and Medicare taxes. A little reflection shows that the Social Security and Medicare systems, are relatively unattractive when considered from the viewpoint of retirement programs.

4) The previous examples overlooked tax considerations. Suppose a married couple earns \$42,000 after taxes. The couple expects their yearly income, saving, and spending to grow 1% over inflation throughout the next sixty years. They plan to put \$3000 yearly in a Roth IRA devoted to an S&P 500 Index fund which earned an average return of 18.6% yearly over the last 10 years. If a conservative return of 10% is anticipated during 40 working years and 20 retirement years, what pension amount can be expected at time of retirement? Take $s = 3,\ R = 10,\ A = 5,\ I = 4,\ t_1 = 40,\ t_2 = 20.$ This leads to $r = 6,\ a = 1,\ u = 5$ so that $p = 3(e^{(.05)(40)}-1)/(1-e^{-(.05)(20)}) = \$30.322$ thousand per year. Since the investment growth will exceed inflation by 1%, the pension value at retirement will be $(30.322)(e^{(.01)(40)}) = \$45.235$ thousand per year and will increase 5% yearly for 20 years. Consequently, the initial pension upon retirement will significantly exceed the initial living expense amount of $42-3=\$39$ thousand but will be significantly less than the living expense amount of \$58.181 thousand available immediately before retirement.

One further observation is appropriate regarding investments made periodically, such as by monthly payroll deductions. When the investment return R is compounded periodically, say k times per year rather than continuously, an equivalent continuous approximation can be obtained as $R \leftarrow k\log(1+R/k)$. Such a calculation is probably unnecessary for financial projections where the expected rate of return is itself approximated. For instance, 10% compounded monthly compares with a continuously compounded rate of 9.96%.

Numerical Analysis

A MATLAB program was written to evaluate the above equations for arbitrary values of the parameters q_0, R, A, I, s, p, t_1 and t_2. The quantity $q(t)$ is plotted for $0 \le t \le (t_1 + t_2)$. The program listing and graphical output for a data case where

$R = 11,\ A = 4,\ I = 4,\ s = 4,\ q_0 = 10,\ q_2 = 100,\ t_1 = 40,\ t_2 = 20$

appears below. Data input for the program is interactive when savspndr is called without any input variables. A call using only one input variable, such as savspndr(0), produces results for the data case just mentioned

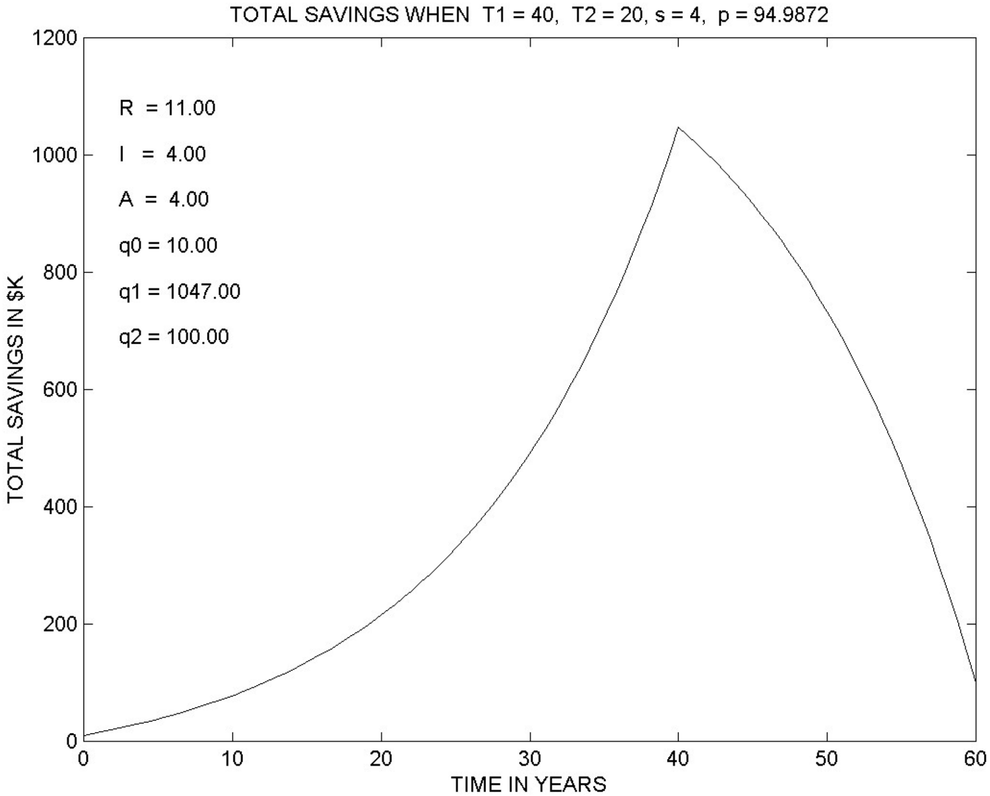
```
savspndr;

    ANALYZING THE SAVE-SPEND PROBLEM BY SOLVING
q'(t)=(R-I)*q+[s-(s+p)*(t>t1)]*exp((A-I)*t, q(0)=q0


Input the percent return on investment (R), the
percent rate of inflation (I), and the percent
rate at which saving and payout are increased to
offset inflation. (Typical values are 11,4,4)
? 11,4,4

Input the number of saving years (t1) and the number
of payout years (t2). (Typical values are 40,20)
? 40,20

Input four values defining saving rate (s), payout rate (p),
initial savings (q0) and final savings (q2). Use nan for one
of these values which will be computed from the other three
(Typical s,p,q0,q2 values in thousands could be 4,nan,10,100)
? 4,nan,10,100
```



Program Source Code

```
function [q,t,s,p,q0,q1,q2]=savspndr...
                          (R,A,I,t1,t2,s,p,q0,q2)
% [q,t,s,p,q0,q1,q2]=savspndr...
%                         (R,A,I,t1,t2,s,p,q0,q2)
% This function solves the SAVE-SPEND PROBLEM
% where funds earning interest are accumulated
% during one period and consumed in a subsequent
% period. The value of assets is adjusted to
% account for effects of inflation. This problem
% is governed by the differential equation
% q'(t)=(R-I)*q(t)-[s-(s+p)*(t>t1)]*exp((A-I)*t)
% where the parameters are defined below
%
% R  - annual percent earnings on assets
% I  - annual percent rate of inflation
% A  - annual percent at which savings are
%      increased to compensate for inflation
% t1 - length in years of the saving
%      period where 0 < t < t1
% t2 - length in years of the payout
%      period where t1 < t < (t1+t2)
% s  - saving multiplier ( $K per year) in the
%      saving rate which is
%      s*exp((A-I)*t),  0<t<t1
% p  - payout multiplier ($K per year) for the
%      period when savings are consumed so the
%      saving rate is negative and equal to
```

```
%       -p*exp((A-I)*t), t1<t<(t1+t2)
% q0 - initial value of savings ($K)
% q2 - final value of savings at t=t1+t2, ($K)
%
% The data input mode depends on MATLAB variable
% nargin. When nargin equals one, a default data
% case is used. When nargin is zero, interactive
% input is used. Otherwise the nine parameters
% in the argument list should be given.
%
% q  - vector of total inflation adjusted
%       savings for 0<t<(t1+t2)
% t  - vector of times (in years) corresponding
%       to the components of q
% q1 - the value of savings at t=t1
%
% NOTE: WHEN R,I,A,T1,T2 ARE KNOWN,THEN FIXING
% ANY THREE OF THE VALUES q0,s,p,q2 DETERMINES
% THE UNKNOWN VALUE WHICH SHOULD BE GIVEN AS
% nan IN THE DATA INPUT. IF ERRONEOUS DATA
% SPECIFYING ALL FOUR VALUES IS USED, THEN A
% CORRECT VALUE OF q2 IS COMPUTED IN TERMS OF
% q0,s, AND p.

%----------------------------------------------
disp(' '), disp(['     ',...
'ANALYZING THE SAVE-SPEND PROBLEM BY SOLVING'])
disp(...
['q''(t)=(R-I)*q+[s-(s+p)*(t>t1)]*',...
 'exp((A-I)*t, q(0)=q0']), disp(' ')

% Default data case
if nargin==1

  R=11; A=4; I=4; t1=40; t2=20;
  s=4; p=nan; q0=10; q2=100;

% Read data interactively
elseif nargin==0

    disp(['Input the percent return on ',...
        'investment (R), the'] )
  disp(['percent rate of inflation (I), ',...,
        'and the percent'])
  disp(['rate at which saving and payout are',...
        ' increased to'])
  disp(['offset inflation. ',...
        '(Typical values are 11,4,4)'])
  [R,I,A]=read;  disp(' ')
  disp(['Input the number of saving years ',...
        '(t1) and the number'])
  disp(['of payout years (t2). (Typical ',...
        'values are 40,20)'])
  [t1,t2]=read; disp(' ')
  disp(['Input four values defining saving ',...
       'rate (s), payout rate (p),'])
  disp(['initial savings (q0) and final ',...
```

```
        'savings (q2). Use nan for one'])
  disp(['of these values which will be',...
         ' computed from the other three'])
  disp(['(Typical s,p,q0,q2 values in',...
        ' thousands could be 4,nan,10,100)'])
  [s,p,q0,q2]=read;

else

  if nargin<9, q2=0; end; if nargin<8, q0=0; end

end

% Check to see whether data is correct
snan=sum(isnan(s)+isnan(p)+isnan(q0)+isnan(q2));
if snan>1
  disp(...
     'DATA ERROR. ONE AND ONLY ONE VALUE AMONG')
  disp('THE PARAMETERS s,p,q0,q2 CAN EQUAL nan')
  return
elseif snan==0
  q2=nan;
end

[q,t,s,p,q0,q1,q2]=savespnd(...
                         R,A,I,t1,t2,s,p,q0,q2);

lab1=sprintf('R  = %5.2f',R);
lab2=sprintf('I   = %5.2f',I);
lab3=sprintf('A  = %5.2f',A);
lab4=sprintf('q0 = %5.2f',q0);
lab5=sprintf('q1 = %5.2f',q1);
lab6=sprintf('q2 = %5.2f',q2);

plot(t,q,'k')
title(['INFLATION ADJUSTED SAVINGS WHEN ',...
 'S = ',num2str(s),' AND P = ',num2str(p)]);
titl=...
['TOTAL SAVINGS WHEN  T1 = ',num2str(t1),...
',  T2 = ',num2str(t2),', s = ',num2str(s),...
',  p = ',num2str(p)]; title(titl)

xlabel('TIME IN YEARS')
ylabel('TOTAL SAVINGS IN $K')
w=axis; ymin=w(3); dy=w(4)-w(3);
xmin=w(1); dx=w(2)-w(1);
ytop=ymin+.9*dy; Dy=.065*dy;
xlft=xmin+0.04*dx;
text(xlft, ytop, lab1)
text(xlft, ytop-Dy, lab2)
text(xlft, ytop-2*Dy, lab3)
text(xlft, ytop-3*Dy, lab4)
text(xlft, ytop-4*Dy, lab5)
text(xlft, ytop-5*Dy, lab6)

grid off,  shg
```

```
%=============================================

function [q,t,s,p,q0,q1,q2]=savespnd...
                        (R,A,I,t1,t2,s,p,q0,q2)
% [q,t,s,p,q0,q1,q2]=savespnd...
%                       (R,A,I,t1,t2,s,p,q0,q2)
% See function savspndr for further discussion
% of the variables in this problem
% R  - annual percent earnings on assets
% I  - annual percent rate of inflation
% A  - annual percent at which savings are
%      increased to compensate for inflation
% t1 - length in years of the saving
%      period where 0 < t < t1
% t2 - length in years of the payout
%      period where t1 < t < (t1+t2)
% s  - saving multiplier ( $K per year) in the
%      saving rate which is
%      s*exp((A-I)*t),  0<t<t1
% p  - payout multiplier ( $K per year) for the
%      period when savings are consumed so the
%      saving rate is negative and equal to
%      -p*exp((A-I)*t), t1<t<(t1+t2)
% q0 - initial value of savings ($K)
% q2 - final value of savings at t=t1+t2, ($K)
%
% q  - vector of total inflation adjusted
%      savings ($K)  for 0<t<(t1+t2)
% t  - vector of times (in years) corresponding
%      to the components of q
% q1 - the value of savings ($K) at t=t1

r=(R-I)/100; a=(A-I)/100;

% If r equals a, shift parameter a tiny amount
if a==r; a=r*(1+1e-6); end; b=r-a;
if t1==0, t1=t2/1e6; end;
if t2==0, t2=t1/1e6; end

% Linear equation relating s,p,q2 and q0
% c1*s-c2*p=c3*q2-c4*q0;
c1=exp(b*t1)-1; c2=1-exp(-b*t2);
c3=b*exp(-a*t1-r*t2); c4=c3*exp(r*t1+r*t2);

% Solve for the one parameter which is unknown
if isnan(s)
  s=(c2*p+c3*q2-c4*q0)/c1;
elseif isnan(p)
  p=(c1*s-c3*q2+c4*q0)/c2;
elseif isnan(q2)
  q2=(c1*s-c2*p+c4*q0)/c3;
else
  q0=(-c1*s+c2*p+c3*q2)/c4;
end

% Savings at t=t1
q1=q0*exp(r*t1)+s/b*(exp(r*t1)-exp(a*t1));
```

```
% Compute vectors for of t and q
n=max(1,round(100*t1/(t1+t2))); m=99-n;
t=[[t1/n*(0:n)],[t1+(t2/m*(1:m))]];
q=q0*exp(r*t)+s/b*(exp(r*t)-exp(a*t))...
  -(s+p)/b*exp(a*t1)*(t>t1).*(...
  exp(r*(t-t1))-exp(a*(t-t1)));

%==============================================

function [a1,a2,a3,a4,a5,a6,a7,a8,a9,a10, ...
          a11,a12,a13,a14,a15,a16,a17,a18, ...
          a19,a20]=read(labl)
%
% [a1,a2,a3,a4,a5,a6,a7,a8,a9,a10,a11,a12, ...
%  a13,a14,a15,a16,a17,a18,a19,a20]=read(labl)
%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
%
% This function reads up to 20 variables on one
% line. The items should be separated by commas
% or blanks. Using more than 20 output
% variables will result in an error.
%
% labl                - Label preceding the
%                       data entry.  It is set
%                       to '? ' if no value of
%                       labl is given.
% a1,a2,...,a_nargout - The output variables
%                       which are created
%                       (cannot exceed 20)
%
% A typical function call is:
% [A,B,C,D]=read('Enter values of A,B,C,D: ')
%
% User m functions required: none
%----------------------------------------------

if nargin==0, labl='? '; end; n=nargout;
str=input(labl,'s'); str=['[',str,']'];
v=eval(str); L=length(v);
if L>=n
  v=v(1:n);
else
  v=[v,zeros(1,n-L)];
end
for j=1:nargout
 eval(['a',int2str(j),'=v(j);']);
end
```