```
function [a,b,yfit] = Fseries(x,y,n,scale,sincos)
% FSERIES Computes real Fourier series approximation to a data set
%
% [A,B] = FSERIES(X,Y,N) fits an Nth order Fourier expansion of the form
%    y = A_0/2 + Sum_k[ A_k cos(kx) + B_k sin(kx) ]
% to the data in the vectors X & Y, using a least-squares fit.
%
% [A,B,YFIT] = FSERIES(X,Y,N) also returns a vector YFIT of the y values
% obtained by evaluating the fitted model at the values in X.
%
% [A,B,YFIT] = FSERIES(X,Y,N,RESCALING) scales the X data to lie in the
% interval [-pi,pi] if RESCALING is TRUE (default).  If RESCALING is
% FALSE, no rescaling of X is performed.
%
% [A,B,YFIT] = FSERIES(X,Y,N,RESCALING,TYPE) uses a sine expansion if TYPE
% is 'sin' or a cosine expansion if TYPE is 'cos'.  Both A & B are still
% returned, however A will be all zero if TYPE = 'sin' and B will be all
% zero if TYPE = 'cos'.
%
% See also: Fseriesval


if nargin<3
   error('MATLAB:Fseries:MissingInputs','Required inputs are x, y, and n')
end
checkinputs();


% scale x to [-pi,pi]
```

```
if scale
   x1 = min(x);
   x2 = max(x);
   x = pi*(2*(x-x1)/(x2-x1) - 1);
end

% make design matrix
nx = x*(1:n);
if isequal(sincos,'b')
   F = [0.5*ones(size(x)),cos(nx),sin(nx)];
elseif isequal(sincos,'c')
   F = [0.5*ones(size(x)),cos(nx)];
else
   F = sin(nx);
end

% do fit
c = F\y;

% extract coefficients
if isequal(sincos,'b')
   a = c(1:n+1);
   b = c(n+2:end);
elseif isequal(sincos,'c')
   a = c;
   b = zeros(n,1);
else
   a = zeros(n+1,1);
```

```
    b = c;
end

% evaluate fit
yfit = F*c;

% transpose yfit back to a row, if y was a row
if xrow
    yfit = yfit';
end



    function checkinputs
        % x & y values
        if isnumeric(x) && isvector(x) && isnumeric(y) && isvector(y)
            if ~isequal(size(x),size(y))
                throwAsCaller(MException('MATLAB:Fseries:UnequalData','x and y must be same size'))
            end
            % transpose x & y to columns if they are rows
            if size(x,2)>1
                x = x';
                y = y';
                xrow = true;
            else
                xrow = false;
            end
            % remove anything not finite and real
            idx = ~(isfinite(x) & isfinite(y) & isreal(x) & isreal(y));
```

```
        x(idx) = [];
        y(idx) = [];
        if isempty(x)
            throwAsCaller(MException('MATLAB:Fseries:NoData','x and y contain no real, finite data'))
        end
    else
        throwAsCaller(MException('MATLAB:Fseries:WrongDataType','x and y values must be numeric vectors'))
    end
    % number of terms
    if isscalar(n) && isnumeric(n)
        if n<0
            throwAsCaller(MException('MATLAB:Fseries:NegativeTerms','Number of terms (n) cannot be negative!'));
        end
    else
        throwAsCaller(MException('MATLAB:Fseries:WrongDataType','n must be a numeric scalar'))
    end
    % optional scaling argument
    if exist('scale','var')
        if ~islogical(scale)
            throwAsCaller(MException('MATLAB:Fseries:WrongDataType','Scaling parameter must be logical (true/false)'))
        end
    else
        scale = true;
    end
    % optional type argument
    if exist('sincos','var')
```

```
        if ischar(sincos)
            sincos = lower(sincos);
            if ismember(sincos,{'s','sin','sine'})
                sincos = 's';
            elseif ismember(sincos,{'c','cos','cosine'})
                sincos = 'c';
            else
                throwAsCaller(MException('MATLAB:Fseries:WrongFourierType',['Unknown type:
',sincos]))
            end
        else
            throwAsCaller(MException('MATLAB:Fseries:WrongDataType','Type must be string (''sin'' or
''cos'')'))
        end
    else
        sincos = 'b';
    end
  end


end
```