```
function psi = generate_squeezed_vacuum_vector(varianceOrRatio, maxPhotons, ratioSwitch)
% state vector for squeezed vacuum state in Fock basis
%    generate_squeezed_vacuum_vector(varianceOrRatio, maxPhotons, ...
%        ratioSwitch)
%    returns the state vector in the Fock basis for a squeezed vacuum
%    state.  The level of squeezing is given by varianceOrRatio, which may
%    be the state's x-quadrature variance or the ratio of this variance to
%    the vacuum variance.  ratioSwitch = 'true variance' indicates that
%    varianceOrRatio is the state's variance. ratioSwotch = 'ratio'
%    indicates that varianceOrRatio is ratio of the state's variance with
%    vacuum.  Vacuum variance = 1/2.  The state will be expressed in a
%    Hilbert space with at most maxPhotons.  maxPhotons may also be the
%    struct S generated by init_tables.

vacuumVariance = 0.5;

if exist('ratioSwitch', 'var') && strcmp(ratioSwitch,'true variance')
    variance = varianceOrRatio;
elseif exist('ratioSwitch', 'var') && strcmp(ratioSwitch,'ratio')
    variance = varianceOrRatio*vacuumVariance;
else  % if no ratioSwitch is present, assume the ratio is given
    variance = varianceOrRatio*vacuumVariance;
end

lambda = (variance-vacuumVariance)/(variance+vacuumVariance);

if isstruct(maxPhotons)
    maxPhotons = maxPhotons.photons;
end

% psi will have occupation of only even numbered photons.
% n is the number of nonzero elements of psi
n = floor(maxPhotons / 2);

a=(0:n).';
b=realsqrt(factorial(2*a));
c=factorial(a);
d=(lambda/2).^(a);
psi=b./c.*d;
psi = psi .* ((1 - lambda^2)^(1/4));

psi = normalize(psi, 'check');

% here we add zeros into the odd numbered photons places
psi=[psi.';zeros(1,n+1)];
psi=psi(1:(maxPhotons+1));
psi=psi(:);

end
```