

## 第四题

### Matlab 源码

```
%% test the algorithm
max_n=50;
depature = zeros(1,max_n);
for n = 1:max_n
    U = orth(rand(n,n));
    A = U*diag(10*rand(1,n))*U';
    exp_A_by_function = matrix_exponential(A);
    exp_A = U*exp(diag(1:n))*U';
    depature(1,n) = norm(exp_A_by_function-exp_A,'fro');
end
figure()
plot([1:max_n],log10(depature),'--o');
xlabel("矩阵阶数")
ylabel("log10(depature)")
[t,s] = title('误差（取对数）','矩阵阶数: 1~50')
%% implement the scaling_and_squaring algorithm for computing the matrix
exponential
% (combined with truncated Taylor series)
function exp_M = matrix_exponential(M)
n = size(M);
n = n(1,1);
e = eig(M);
max_eigenvalue = max(e);
k = ceil(log2(max_eigenvalue/0.001));
M_scaled = M/(2^k);
exp_M_scaled = eye(n) + M_scaled + (1/2)*M_scaled*M_scaled +
(1/6)*M_scaled*M_scaled*M_scaled +
(1/24)*M_scaled*M_scaled*M_scaled*M_scaled;% truncated Taylor series
exp_M = exp_M_scaled;
for i = 1:k
    exp_M = exp_M*exp_M;
end
end
```

## 试验结果

