

به نام خدا

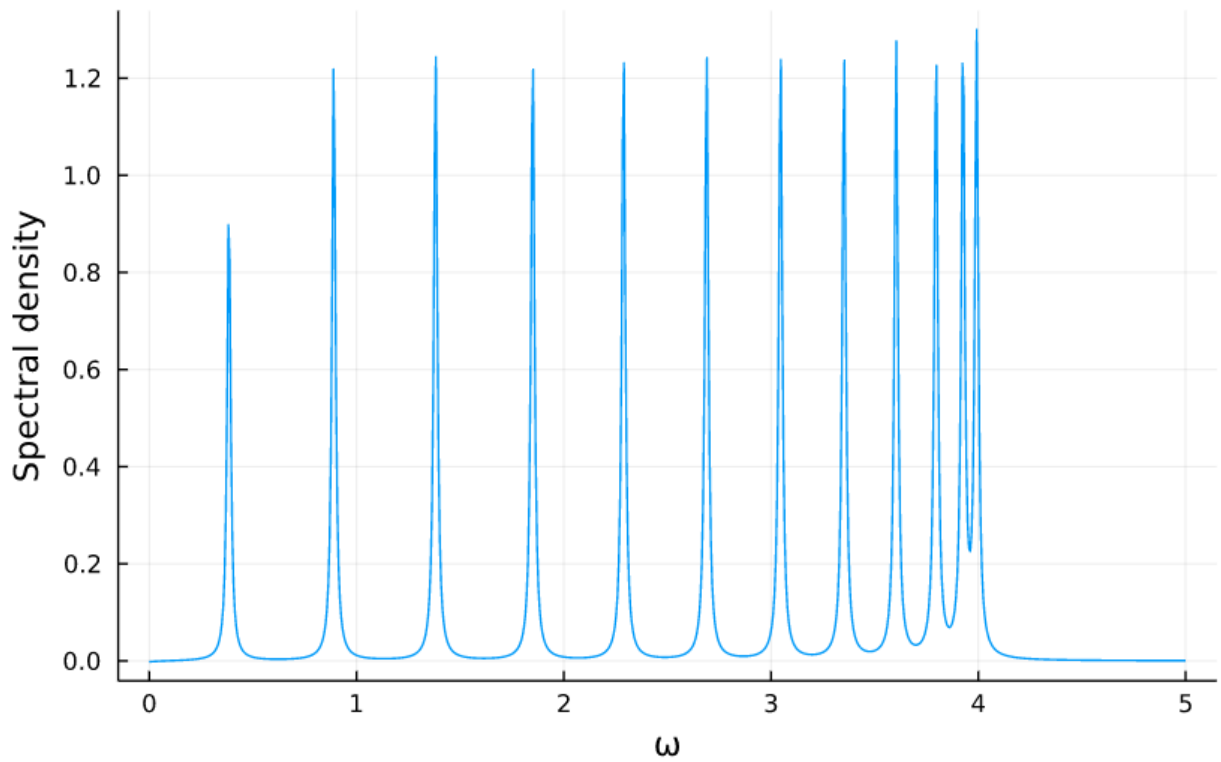
## تمرین سری هفتم: توسعه تحلیلی

زهره فرهمند، یاسمین پناهی، ریحانه آقائی صائم

الف) در این بخش را طبق رابطه‌ای که برای آن به صورت دقیق داریم رسم می‌کنیم تا در ادامه بتوانیم نتایج روش تقریبی را با این جواب دقیق مقایسه کنیم.

$$\Pi(q, iq_n) = \frac{1}{N} \sum_k \frac{n_F(\epsilon_k) - n_F(\epsilon_{k+q})}{iq_n + \epsilon_k - \epsilon_{k+q}}$$

Exact spectral density for 1D electron gas (N = 50)



ب) می‌خواهیم در ادامه از روش تقریبی پاده استفاده کنیم. در کلاس یاد گرفتیم که برای این روش باید ابتدا مجموعه‌ای از زوجی از نقاط را در نظر بگیریم و با داشتن آنها و با حل دستگاه معادلات، ضرایب را پیدا کنیم که با داشتن ضرایب می‌توانیم تابع همبستگی ماتسوبارا را برازش کنیم.

بنابراین مجموعه‌ای  $2m$  تایی از زوج‌های  $\{\Pi(q = \pi, iq_n), iq_n\}$  با دقت  $D$  تولید می‌کنیم.

ج و د) در این بخش با استفاده از زوج داده‌هایی که در بخش قبل به دست آوردیم باید دستگاه معادلاتی که داریم را حل کنیم. این دستگاه معادلات خطی را به صورت ماتریسی می‌نویسیم.

$$\Gamma(z) = \frac{P_m(z)}{Q_m(z)} = \frac{\sum_{j=0}^m p_j z^j}{1 + \sum_{j=1}^m q_j z^j}$$

$$\sum_{j=0}^m p_j z^j - \Gamma(z) \sum_{j=1}^m q_j z^j = \Gamma(z)$$

ماتریس مجهول‌ها را  $x$  می‌گیریم و  $2m + 1$  مجهولی که داریم (که شامل  $q_j$  و  $p_i$  ها هستند) را در آن به ترتیب زیر قرار می‌دهیم.

$$x = \begin{bmatrix} p_0 \\ \vdots \\ p_m \\ q_1 \\ \vdots \\ q_m \end{bmatrix}_{(2m+1) \times 1} \quad y = \begin{bmatrix} \Gamma(z^{(1)}) \\ \Gamma(z^{(2)}) \\ \vdots \\ \Gamma(z^{(2m+1)}) \end{bmatrix}_{(2m+1) \times 1}$$

$$M = \begin{bmatrix} 1 & z^{(1)} & (z^{(1)})^2 & \dots & (z^{(1)})^m & -\Gamma(z^{(1)})z^{(1)} & -\Gamma(z^{(1)})(z^{(1)})^2 & \dots & -\Gamma(z^{(1)})(z^{(1)})^m \\ & \vdots & & \ddots & & & \vdots & & \\ 1 & z^{(n)} & (z^{(n)})^2 & \dots & (z^{(n)})^m & -\Gamma(z^{(n)})z^{(n)} & -\Gamma(z^{(n)})(z^{(n)})^2 & \dots & -\Gamma(z^{(n)})(z^{(n)})^m \end{bmatrix}_{(2m+1) \times (2m+1)}$$

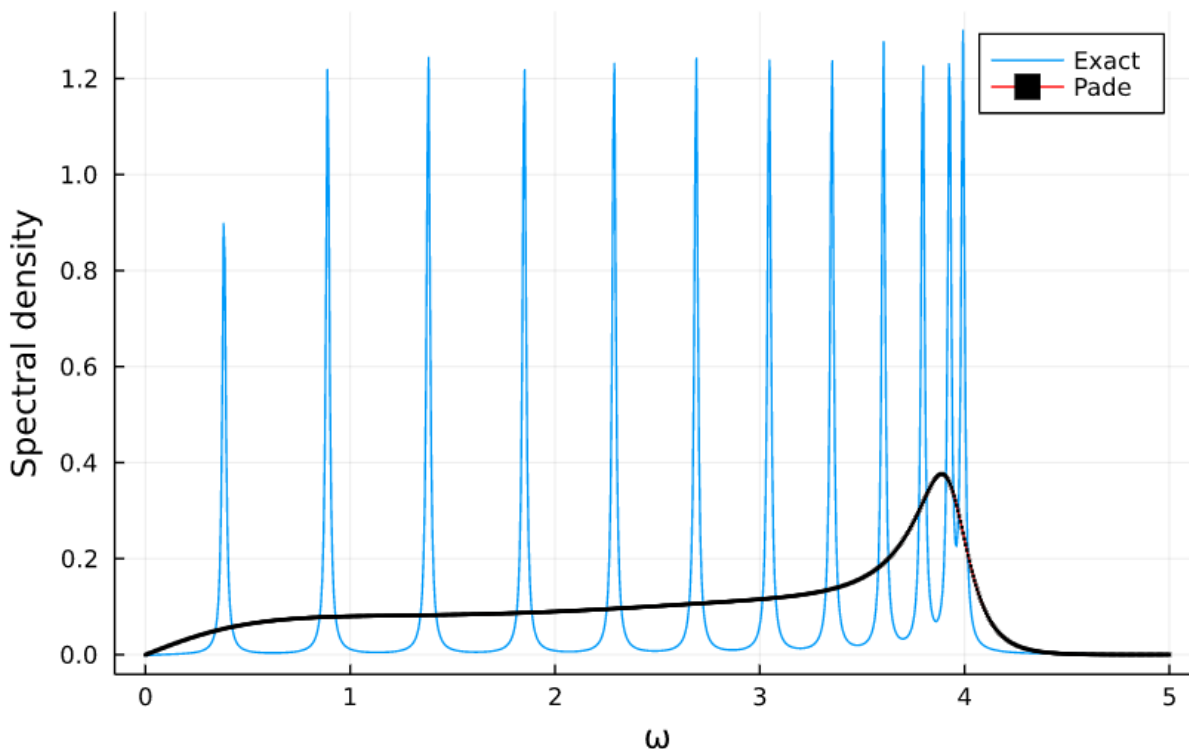
که در رابطه بالا  $n = 2m + 1$  است. و در نهایت دستگاه معادلات خطی به صورت زیر حل خواهد شد.

$$M x = y$$

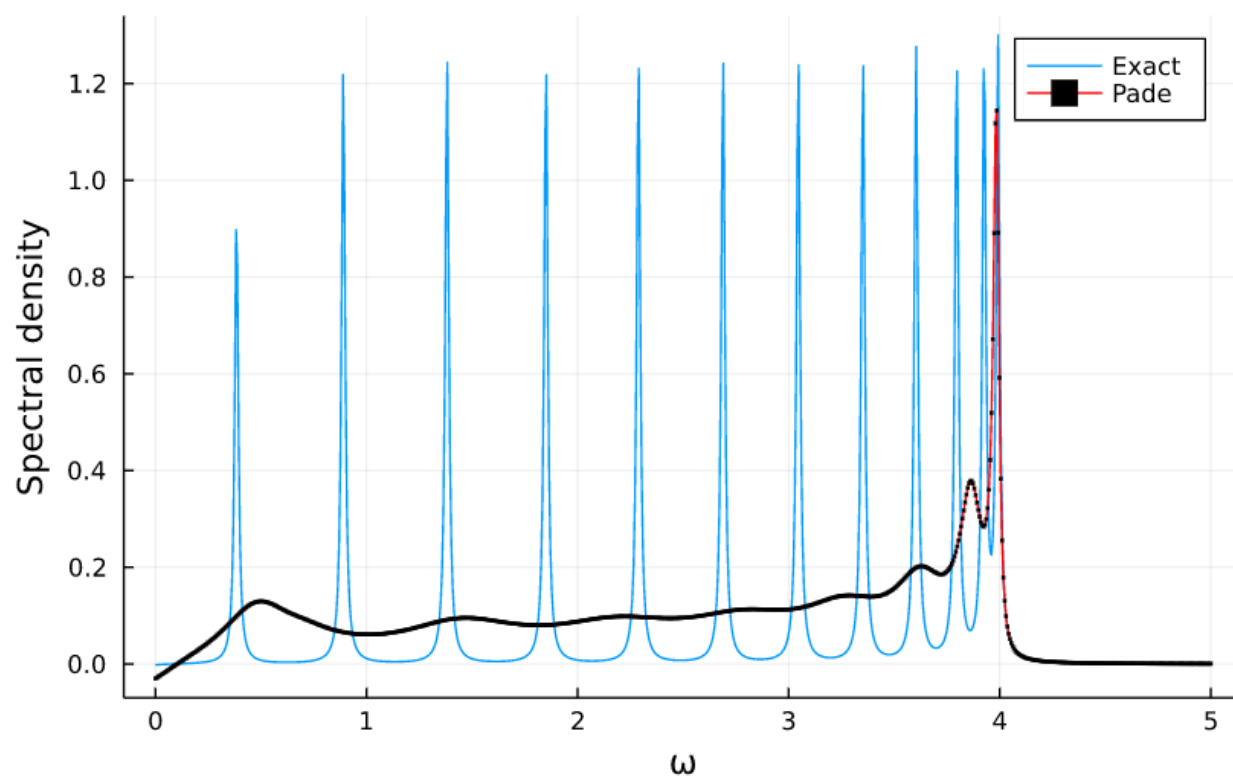
$$x = M^{-1} y$$

نمودارهای مربوط به دقت های  $D = 16, 56, 65, 82$  در ادامه آورده شده است.

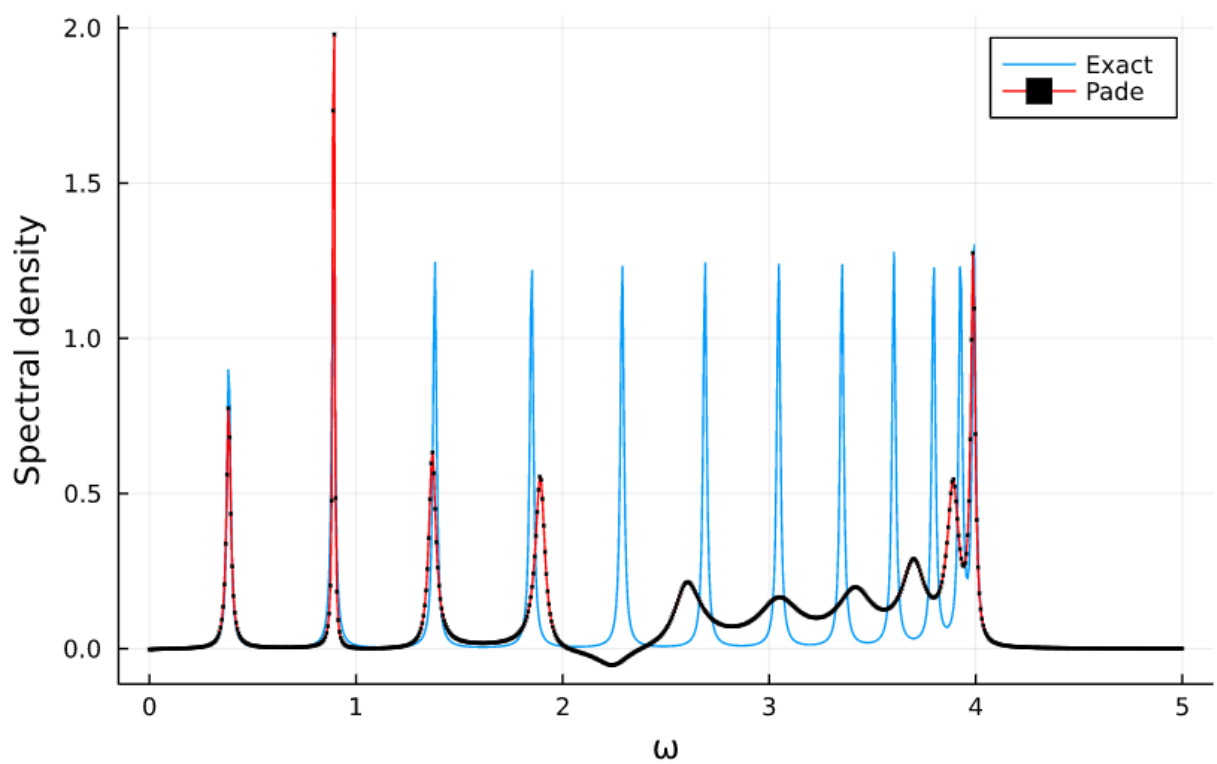
### Pade Approx. (D = 16) for 1D electron gas (N = 50)



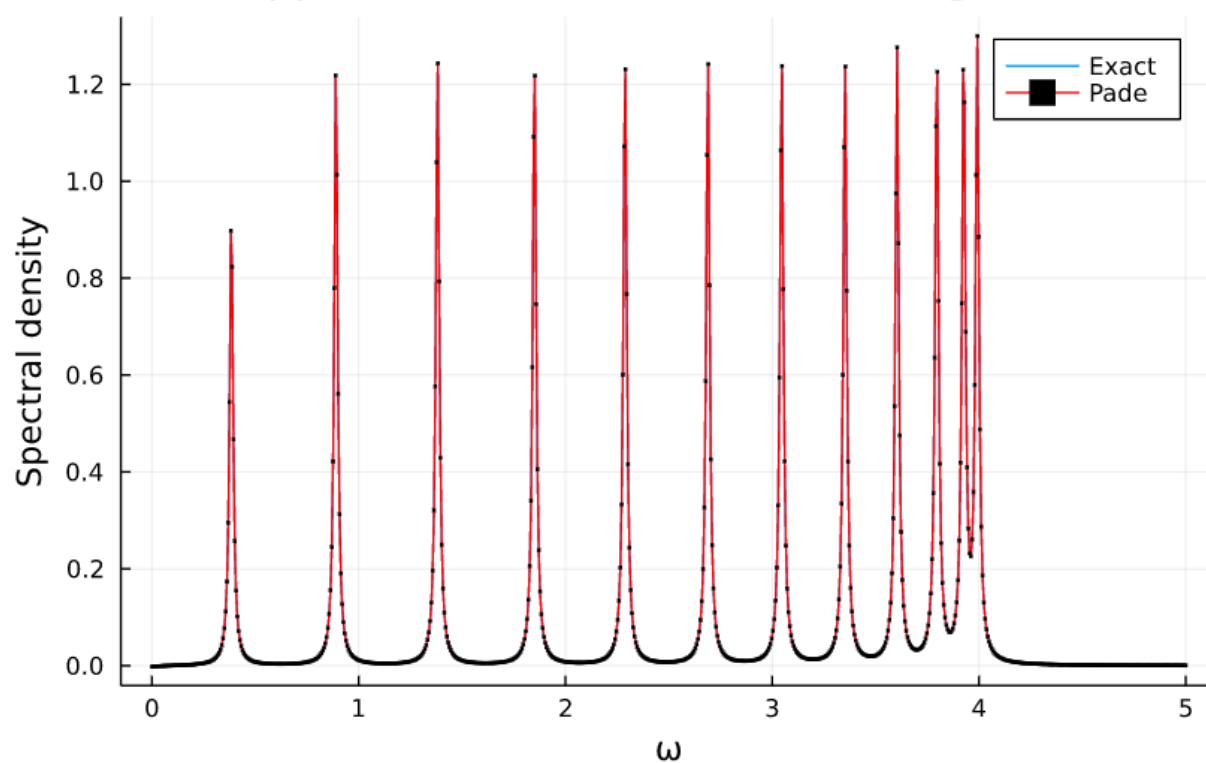
Pade Approx. ( $D = 56$ ) for 1D electron gas ( $N = 50$ )



Pade Approx. ( $D = 65$ ) for 1D electron gas ( $N = 50$ )



### Pade Approx. ( $D = 82$ ) for 1D electron gas ( $N = 50$ )



در ادامه گزارش کد به زبان جولیا آمده است. برای حل این سوال از متلب و پایتون هم استفاده کردیم اما با تعیین کردن دقت در متلب و پایتون مشکلاتی داشتیم که در نهایت توانستیم این مشکلات را در جولیا برطرف کنیم.

```
In [287]: import Pkg;
# Pkg.add("BenchmarkTools");
using BenchmarkTools;
using LinearAlgebra;
# Pkg.add("DataStructures");
using DataStructures;
# Pkg.add("StatsBase");
using StatsBase;
# Pkg.add("Arpack");
using Arpack;
# Pkg.add("Plots")
using Plots;
# Pkg.add("Statistics");
using Statistics;
# Pkg.add("Plots");
using Plots;
# Pkg.add("LaTeXStrings");
using LaTeXStrings;
```

```
In [288]: # for each precision we should first set it at the begginig of the code
setprecision(219)
```

Out[288]: 219

```
In [289]: function  $\Pi$ _func(q, qn, k, t,  $\beta$ ,  $\mu$ , N)

    E_k = Energy(k, t);
    E_kq = Energy(k .+ q, t);

    nF_k = FermiDirac(E_k,  $\beta$ ,  $\mu$ );
    nF_kq = FermiDirac(E_kq,  $\beta$ ,  $\mu$ );

     $\Pi$ _q = 1/N * sum( (nF_k .- nF_kq) ./ (qn .+ E_k .- E_kq) );

    return  $\Pi$ _q

end
```

Out[289]:  $\Pi$ \_func (generic function with 1 method)

```
In [290]: function Energy(k, t)

    E_k = - 2*t * BigFloat(cos.(k));
    return E_k

end
```

Out[290]: Energy (generic function with 1 method)

```
In [291]: function FermiDirac(E_k,  $\beta$ ,  $\mu$ )

    nF_k = BigFloat.(1 ./ (1 .+ BigFloat.(exp( $\beta$  * (E_k .-  $\mu$ )))));

    end
```

Out[291]: FermiDirac (generic function with 1 method)

# Main

```
In [292]:  $\beta$  = BigFloat(10);  
 $\eta$  = BigFloat(0.01);  
 $\mu$  = BigFloat(0);  
 $t$  = BigFloat(1);  
N = 50;  
L = 800;  
k = BigFloat.(LinRange(-BigFloat( $\pi$ ), BigFloat( $\pi$ ), N));  
 $\omega$  = BigFloat.(LinRange(BigFloat(0), BigFloat(5), L));
```

```
In [293]: #  $\Pi$ _EXACT = zeros(ComplexF64, L,1);  
  
 $\Pi$ _EXACT = [ $\Pi$ _func( $\pi$ ,  $\omega[i] + 1im * \eta$ , k, t,  $\beta$ ,  $\mu$ , N) for i=1:L];
```

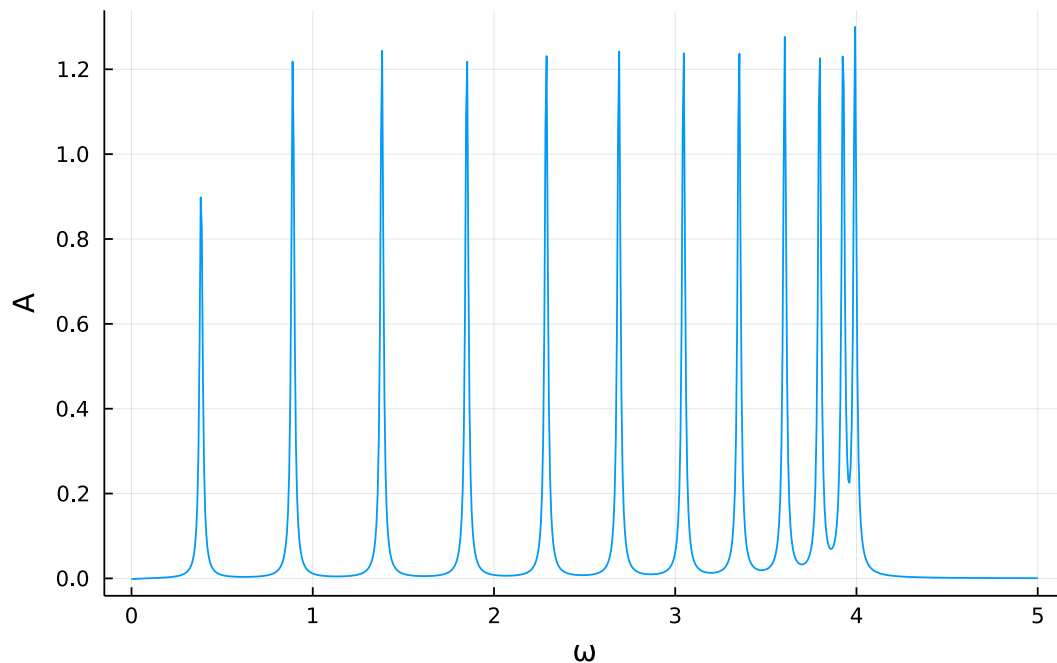
```
In [294]: A_array = -1/BigFloat( $\pi$ ) .* imag( $\Pi$ _EXACT); # spectral density
```

```
In [129]: gr()
```

```
Out[129]: Plots.GRBackend()
```

```
In [296]: plot( $\omega$ , A_array, label = false)  
title!("Spectral density A( $q=\pi$ ,  $\omega$ ) Exact fermionic solution")  
xlabel!(" $\omega$ ")  
ylabel!("A")
```

Out[296]: Spectral density A( $q=\pi$ ,  $\omega$ ) Exact fermionic solution



```
In [297]: m = 30; #Matsubara points = (2m+1)  
qn = 1im * BigFloat( $\pi$ )/ $\beta$  .* ((1:2:(4*m+1)) |> collect);  
n_s = length(qn);  
  
Gamma_z = [ $\Pi$ _func( $\pi$ , qn[i], k, t,  $\beta$ ,  $\mu$ , N) for i=1:n_s]  
  
M = [j <= (m+1) ? qn[i]^(j-1) : (- Gamma_z[i] * qn[i]^(j-(m+1)))] for i=1:n_s, j=1:n_s];
```

```
In [298]: Coef = M\ Gamma_z;

p_coefs = Coef[1:(m+1)];
q_coefs = Coef[(m+2):end];
```

```
In [299]: Gamma_AC = [];

for j = 1:L
    Z = [(w[j] + 1im * η) .^ i for i=1:m];
    Numerator = sum(p_coefs[2:end] .* Z) + p_coefs[1];
    Denominator = sum(q_coefs .* Z) + 1;
    push!(Gamma_AC, Numerator/Denominator)
end
```

```
In [300]: A_pade = -1/BigFloat(π) .* imag.(Gamma_AC);
```

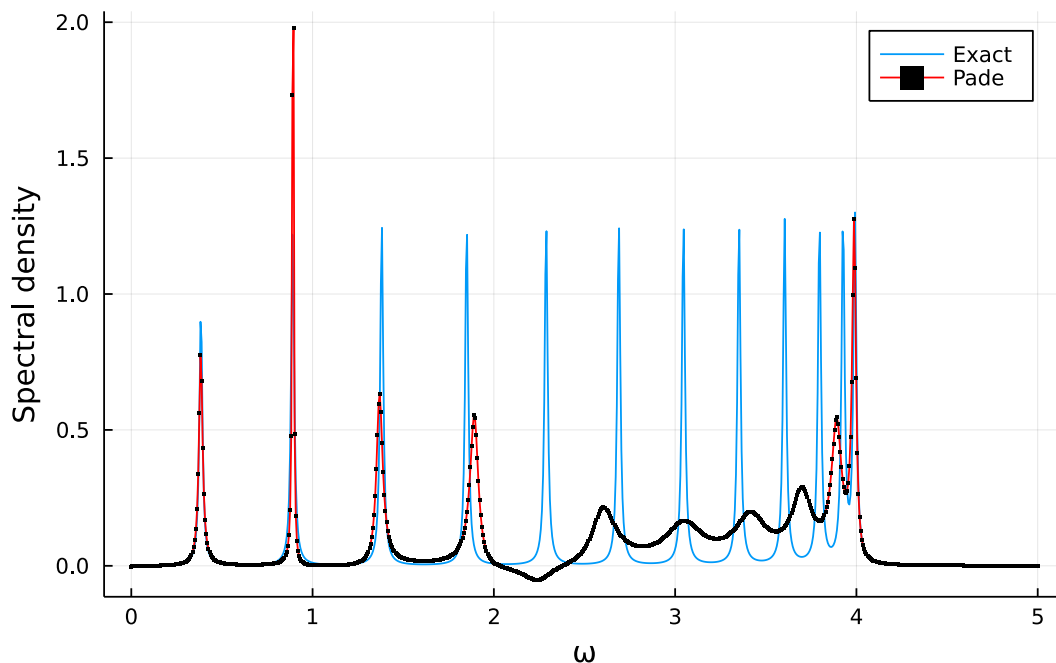
```
In [235]: gr()
```

```
Out[235]: Plots.GRBackend()
```

```
In [302]: L1 = plot(w, A_array, label = "Exact")
L2 = plot!(w, A_pade, linecolor = "red", fillcolor = "black", markercolor = "black", markershape = :auto, markersize = 0.9, label = "Pade")

title!("Pade Approx. (D = 65) for 1D electron gas (N = 50)", fontsize = 8)
xlabel!("ω")
ylabel!("Spectral density")
```

```
Out[302]: Pade Approx. (D = 65) for 1D electron gas (N = 50)
```



```
In [ ]:
```