Computing Padé approximants

```
>> syms x a b c d e
>> T = taylor(exp(x), x, 0, 'Order', 5)
T =
x^4/24 + x^3/6 + x^2/2 + x + 1
>> D = x^2+a*x+b;
>> N = c*x^2 + d*x + e;
>> collect(expand(T*D-N))
ans =
x^6/24 + (a/24 + 1/6)*x^5 + (a/6 + b/24 + 1/2)*x^4 + (a/2 +
>> C = coeffs(collect(expand(T*D-N)),x)
C =
[ b - e, a + b - d, a + b/2 - c + 1, a/2 + b/6 + 1, a/6 + b
>> S = solve(C(1:5),[a,b,c,d,e]);
>> [S.a, S.b, S.c, S.d, S.e]
ans =
[ -6, 12, 1, 6, 12]
```

Accuracy of Padé approximants

```
>> ezplot(exp(x), -5, 5);
>> hold on;
>> ezplot(pade(exp(x), x, 'Order', [2,2]), -5, 5);
```

We saw that $D(A)^{-1}N(A) = \exp(A + H)$, where $H = f(A)$ corresponds to the matrix function $f(x) = \log(\exp(-x)\frac{N(x)}{D(x)})$

```
>> P = pade(exp(x), x, 'Order', [2,2]);
>> T = taylor(log(exp(-x)*P), 'Order', 20)
T =
- x^19/98035826688 - x^17/7309688832 + x^13/38817792 + x^1
>> AbsC = abs(coeffs(T,'All'))
AbsC =
[ 1/98035826688, 0, 1/7309688832, 0, 0, 0, 1/38817792, 0, 
>> % Solve |C|(x) = 2e-16
>> double(solve(C * x.^transpose(19:-1:0) - 2e-16, x, 'Rea
ans =
   3.1037e-03
```

Remark

The same technique (both to construct Padé approximants and to evaluate their backward stability) can be applied to other functions as well; the exponential is just a nice example.

In general, these rational approximations work well only when the eigenvalues are in a sufficiently small region.