*Example 1.12 (*NACLAB *demo:* LINEARSOLVE*).* Suppose, with the data error bound 10^{-4}, the polynomials

$$\begin{aligned} \tilde{f}(x) &= 5.99999 - 9x + 3x^3 - 4x^4 + 6x^5 - 2x^7 \\ \tilde{g}(x) &= -2 + 5x - 3x^2 - x^3 + x^4 + 2x^5 - 3.00002x^6 + x^8 \end{aligned}$$

are empirical data of a polynomial pair f and g with a greatest common divisor

$$u \;=\; \mathscr{gcd}(f,g) \;\approx\; 2 - 2.99999x + 1.00001x^3.$$

given approximately. It is known that the greatest common divisor is in the range of the linear transformation

$$\begin{array}{rcccl} L & : & \mathbb{P}_5 \times \mathbb{P}_4 & \longrightarrow & \mathbb{P}_{12} \\ & & (p,q) & \longmapsto & pf + qg \end{array} \tag{1.36}$$

and the kernel

$$\mathscr{Kernel}(L) \;=\; \mathit{span}\left\{ \left(\frac{f}{u}, \frac{g}{u} \right) \right\}$$

where the notation $\mathbb{P}_n$ denotes the vector space of polynomials with degrees up to n. The question: Can we use the empirical data of f, g and u to solve the linear equation

$$p\tilde{f} + q\tilde{g} \;=\; \tilde{u}$$

for $(p,q) \in \mathbb{P}_5 \times \mathbb{P}_4$ approximately with an accuracy comparable to the data accuracy?

NACLAB provides a comprehensive linear equation solver LINEARSOLVE for such problems directly. Instead of constructing the representation matrix for the linear transformation, write a simple Matlab M-file implementing the linear transformation (1.36) with parameters f and g as it is:

```
function z = map2gcd(p,q,f,g)
%
%   The linear transformation (p,q) --> p*f+q*g
%
    z = PolynomialPlus(PolynomialTimes(p,f),PolynomialTimes(q,g));
```

Enter the data in Matlab in WYSIWYG style:

```
>> f = '5.99999 - 9*x + 3*x^3 - 4*x^4 + 6*x^5 - 2*x^7';
>> g = '-2 + 5*x - 3*x^2 - x^3 + x^4 + 2*x^5 - 3.00002*x^6 + x^8';
>> u = '2-2.99999*x+1.00001*x^3';
```

Define the domain of the linear transformation by providing two polynomials in $\mathbb{P}_5$ and $\mathbb{P}_4$ along with the parameter:

```
>> domain = {'1+x+x^2+x^3+x^4+x^5','1+x+x^2+x^3+x^4'};
>> parameter = {f,g};
```

Call the NACLAB module LINEARSOLVE with input items consist of the linear transformation array `{@maps2gcd, domain, parameter}`, the right-hand side `u`, and error tolerance 10^{-4}:

```
>> [Z,K,lcond,res] = LinearSolve({@map2gcd,domain,parameter}, u, 1e-4);
```

The cell array `Z` contains the numerical solution in $\mathbb{P}_5 \times \mathbb{P}_4$ in WYSIWYG style

```
>> Z{:}
ans =

0.315213638159614 + 0.0325895731888634*x + 0.0217037544785598*x^2 +
0.0542602119088428*x^3 + 0.135649516609881*x^4 + 0.0239081327980908*x^5

ans =

-0.054358026714861 + 0.0434095337991092*x + 0.108521146411911*x^2 +
0.271298482527209*x^3 + 0.0478164653722352*x^4
```

The cell array `K` contains the 1-dimensional numerical Kernel in $\mathbb{P}_5 \times \mathbb{P}_4$ in WYSIWYG style

```
>> K{1}{:}
ans =

-0.249999389223655 + 0.25000118970689*x + 0.250002152085867*x^5


ans =

-0.749997620161645 + 0.500002204170052*x^4
```

The numerical solution can be verified using the linear transformation M-file

```
>> h = map2gcd(Z{:},ff,gg);
>> PolynomialClear(h,1e-5)   % clear numerical zero coefficients

ans =

1.99999473025102 - 2.9999948313716*x + 1.00000604534541*x^3
```

The output `lcond` and `res` show the condition number for the linear equation is healthy at 52.1 and the residual is below the error tolerance at about 5.27×10^{-6}.