

UMA PROPOSTA EXPERIMENTAL PARA O ENSINO DE MCR-ALS

Nathan S. Perrud^a, Eduarda B. S. Cruz^b, Natália N. M. Deimling^c e Patrícia Valderrama^{b,d,*}

^aDepartamento de Química, Universidade Tecnológica Federal do Paraná (UTFPR), 87301-900 Campo Mourão – PR, Brasil

^bPrograma de Pós-Graduação em Tecnologia de Alimentos (PPGTA), Universidade Tecnológica Federal do Paraná (UTFPR), 87301-900 Campo Mourão – PR, Brasil

^cPrograma de Pós-Graduação em Ensino de Ciências Humanas, Sociais e da Natureza (PPGEN), Departamento de Química, Universidade Tecnológica Federal do Paraná (UTFPR), 87301-900 Campo Mourão – PR, Brasil

^dPrograma de Pós-Graduação em Inovações Tecnológicas (PPGIT), Departamento de Química, Universidade Tecnológica Federal do Paraná (UTFPR), 87301-900 Campo Mourão – PR, Brasil

Recebido: 21/10/2025; aceito: 06/02/2026; publicado online: 02/03/2026

*e-mail: patriciav@utfpr.edu.br; pativalderrama@gmail.com

Patrícia Valderrama: <https://orcid.org/0000-0002-6680-1260>

Este material suplementar tem por finalidade apresentar um passo a passo para a execução da resolução multivariada de curvas com mínimos quadrados alternantes (MCR-ALS, *multivariate curve resolution alternating least squares*) no software Matlab (MathWorks, EUA).

O primeiro passo após a aquisição espectral das amostras consistiu na importação dos espectros obtidos para o Matlab. Considerando arquivos com extensão “.txt”, utilizou-se o comando: >> load NomeDaAmostra.txt;. Esse procedimento foi repetido para as onze amostras consideradas nesse estudo.

Os arquivos contendo os espectros, na extensão ‘.txt’, podem ser abertos no bloco de notas; permitindo a visualização da estrutura dos dados, organizada em duas colunas numéricas: a primeira corresponde aos valores de comprimento de onda, enquanto a segunda apresenta os respectivos valores de absorbância.

A próxima etapa inclui a obtenção da matriz de dados que será posteriormente analisada por MCR-ALS. Para tal, utilizam-se os comandos dispostos no Quadro 1.

Quadro 1. Comandos para a construção da matriz **X**

```
>> X(1,:)=AM100(:,2)';  
>> X(2,:)=AZ10AM90(:,2)';  
>> X(3,:)=AZ20AM80(:,2)';  
>> X(4,:)=AZ30AM70(:,2)';  
>> X(5,:)=AZ40AM60(:,2)';  
>> X(6,:)=AZ50AM50(:,2)';  
>> X(7,:)=AZ60AM40(:,2)';  
>> X(8,:)=AZ70AM30(:,2)';  
>> X(9,:)=AZ80AM20(:,2)';  
>> X(10,:)=AZ90AM10(:,2)';  
>> X(11,:)=AZ100(:,2)';
```

No Quadro 1, os valores entre parênteses representam, respectivamente, linha e coluna da matriz, separados pela vírgula. Assim, a primeira linha do Quadro 1 (>> **X**(1,:)=AM100(:,2)') deve ser interpretada como: a primeira linha da matriz **X**, contendo todas as suas colunas (**X**(1,:)), será igual a todas as linhas da segunda coluna da amostra que contém 100% de alaranjado de metila (=AM100(:,2))'. O sinal de apóstrofo (') indica que os valores serão transpostos. Com isso, a segunda coluna da amostra am10 deixa de ser uma coluna e passa a ser a primeira linha da matriz **X**.

Para a criação de um vetor com os valores de comprimento de onda, pode-se utilizar a primeira coluna de qualquer uma das amostras, pois os espectros de todas as amostras foram obtidos na mesma faixa espectral. Dessa forma, a primeira coluna das amostras é exatamente igual em todos os casos. Utilizaremos como exemplo a amostra com 100% de alaranjado de metila (AM100): >> nm=AM100(:,1);.

Os espectros obtidos para as misturas de corantes, organizados em matriz **X** (11×580), são apresentados na Figura 1S e os comandos para sua obtenção são apresentados no Quadro 2. A partir do comando *plot*, os espectros da matriz **X** serão exibidos em função dos comprimentos de onda (nm). A letra ‘k’ indica que os espectros serão plotados na cor preta. A letra ‘y’ indica que o espectro da primeira amostra (alaranjado de metila 100%) será plotado na cor amarela. A letra ‘b’ indica que o espectro da última amostra (azul de metileno 100%) será plotado na cor azul. O comando *hold on* indica que os espectros das amostras 1 e 11 serão exibidos no mesmo gráfico que os das demais amostras. Nas duas últimas linhas do Quadro 2, os comandos apresentados permitem inserir nomes nos eixos y e x, respectivamente.

Quadro 2. Comandos para obtenção do gráfico dos espectros

```
>> plot(nm,X,'k');
>> hold on;plot(nm,X(1,:),‘y’);
>> hold on;plot(nm,X(11,:),‘b’);
ylabel(‘Absorbância’);
xlabel(‘Comprimento de onda (nm)’);
```

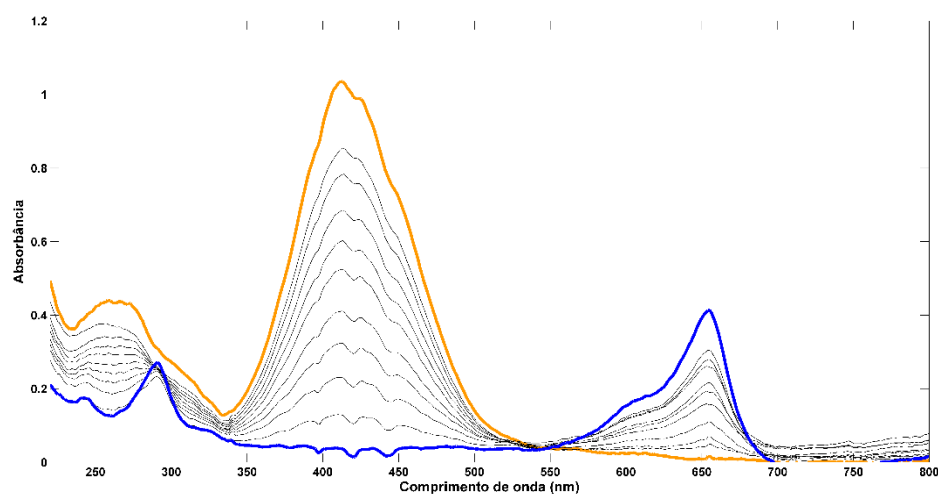


Figura 1S. Espectros UV-Vis das soluções de AZ (—), AM (—) e suas misturas em diferentes proporções (—)

Antes de inicializar o algoritmo MCR-ALS, é necessário salvar todo o trabalho que já foi realizado. Nesse caso, utilizou-se o comando: `>> save mcr.mat`. Na sequência, o comando: `>> mcr-main` foi empregado para abrir a interface gráfica do MCR-ALS, que é apresentada na Figura 2S.

Nessa interface, selecionou-se a matriz **X** previamente organizada; nesse caso, é possível determinar manualmente o número de componentes, que é 2. A estimativa inicial foi realizada por “Pure”, que corresponde ao SIMPLISMA (*simple-to-use interactive self-modeling mixture analysis*) da plataforma ChemFlow.

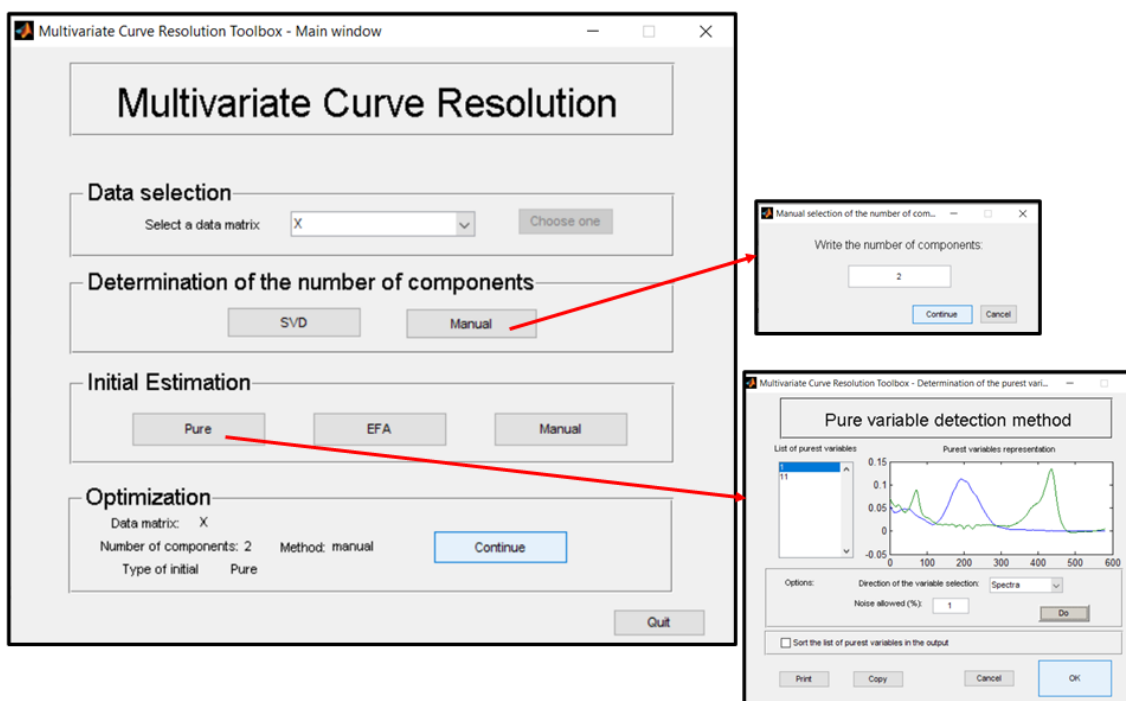


Figura 2S. Interface gráfica do MCR-ALS

Ao inserir o comando “continue”, a interface da Figura 3S será exibida. Nesse caso, indicou-se que o número de matrizes é 1, pois pretende-se aplicar as mesmas restrições a todas as amostras. Se o usuário preferir aplicar restrições distintas às amostras, o número de matrizes deve ser igual ao de amostras. Novamente, o comando “continue” exibirá a próxima interface (Figura 4S), na qual as restrições são impostas antes de iniciar o processo iterativo de otimização.

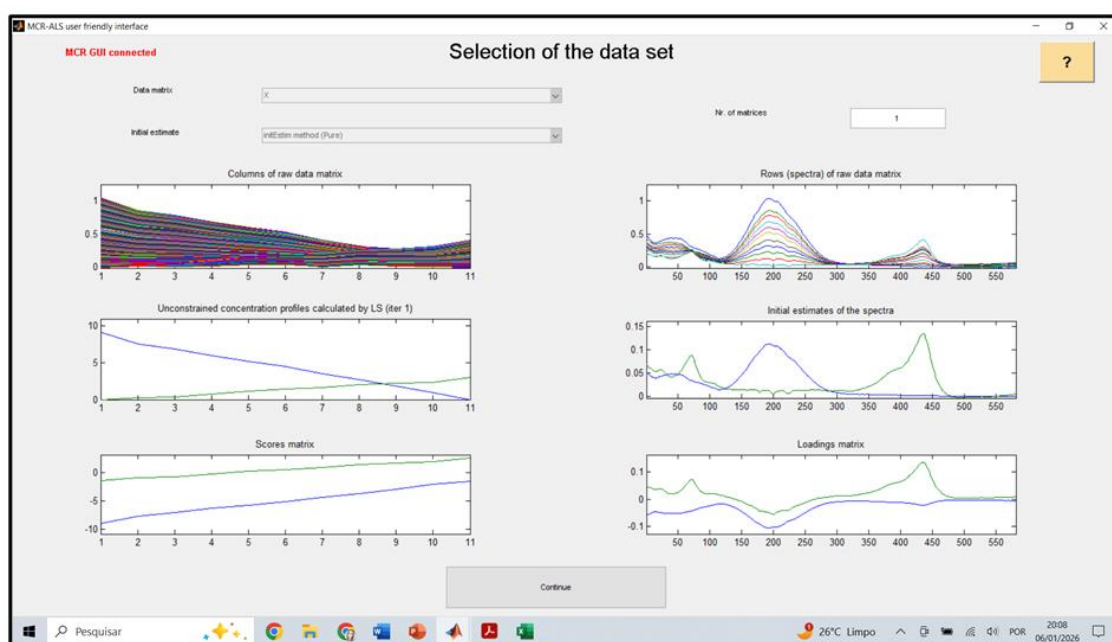


Figura 3S. Interface para indicação do número de matrizes a serem consideradas

MCR-ALS user friendly interface

Selection of ALS constraints

No-negativity
☒ Yes? ☐ Conc ☐ Spectra ☒ Conc & Spec
 Implementation for conc: Implementation for spec:
 Nr. of species with non-neg conc: Enter a vector of positive profiles:
 Nr. of species with non-neg spec: Enter a vector of positive profiles:

Unimodality
☐ Yes? ☐ Conc ☐ Spectra ☐ Conc & Spec
 Implementation of the unimodality constraint:
 Nr. of species with unimodal conc: Species with unimodal conc?:
 Nr. of species with unimodal spec: Species with unimodal spec?:
 Constraint tolerance for conc: Constraint tolerance for spec:

Closure
☐ Yes? ☐ Conc ☐ Spectra
 Nr. of closure constraints to be included?:
 First Closure constant Equal to: Second Closure constant Equal to:
 First variable closure: Second variable closure constants:
 Closure condition: Closure condition:
☐ Closure variable? Which species are in 1st closure? Which species are in 2nd closure?

Equality constraints in conc profiles ☐ Yes? Select osel matrix: Constraints are:
Equality constraints in spectra profiles ☐ Yes? Select ssl matrix: Constraints are:

Optimization parameters
 Nr. of iterations: Convergence criterion: ☐ Graphical output

Output
 Concentration: Std. dev.:
 Spectra: Residuals:
 Area opt: Ratio opt:

Figura 4S. Interface para indicação das restrições

A partir da Figura 4S, existem três possibilidades para a restrição de não-negatividade: (i) *forced to zero*, (ii) *nnls* e (iii) *fnls*. Todas são formas distintas de impor a não negatividade, com diferenças na aplicação da restrição e no impacto na solução.

O primeiro é muito simples e rápido; valores negativos são simplesmente substituídos por zero após um ajuste de mínimos quadrados sem restrições. Porém, não é matematicamente consistente; pode distorcer o ajuste e causar oscilações e falsa convergência. O segundo (*nnls*, *non-negative least squares*) é um método mais rigoroso que resolve o sistema de mínimos quadrados já com a restrição. Com isso, a solução ótima é obtida sob restrição, proporcionando resultados fisicamente coerentes e uma convergência estável. A desvantagem de *nnls* é ser mais lento computacionalmente. Por fim, *fnls* (*fast non-negative least squares*) é a versão otimizada e acelerada de *nnls* e utiliza estratégias de atualização ativa para reduzir o custo computacional, obtendo o mesmo resultado matemático do *nnls* de forma mais rápida.

O número de espécies que não devem apresentar concentrações negativas e espectros recuperados com valores negativos também é escolhido na interface da Figura 4S e, como, nesse caso, considera-se dois componentes (mistura de dois corantes), opta-se por impor as restrições de não negatividade de concentração e de espectro aos 2 constituintes. O número de iterações e o critério de convergência *default* são 50 e 0,1, respectivamente. Entretanto, optou-se por utilizar até 100 iterações, com critério de convergência de 0,001, os mesmos utilizados na plataforma ChemFlow. O próximo passo é otimizar o modelo em “*Optimize*” e obter os resultados conforme a Figura 5S.

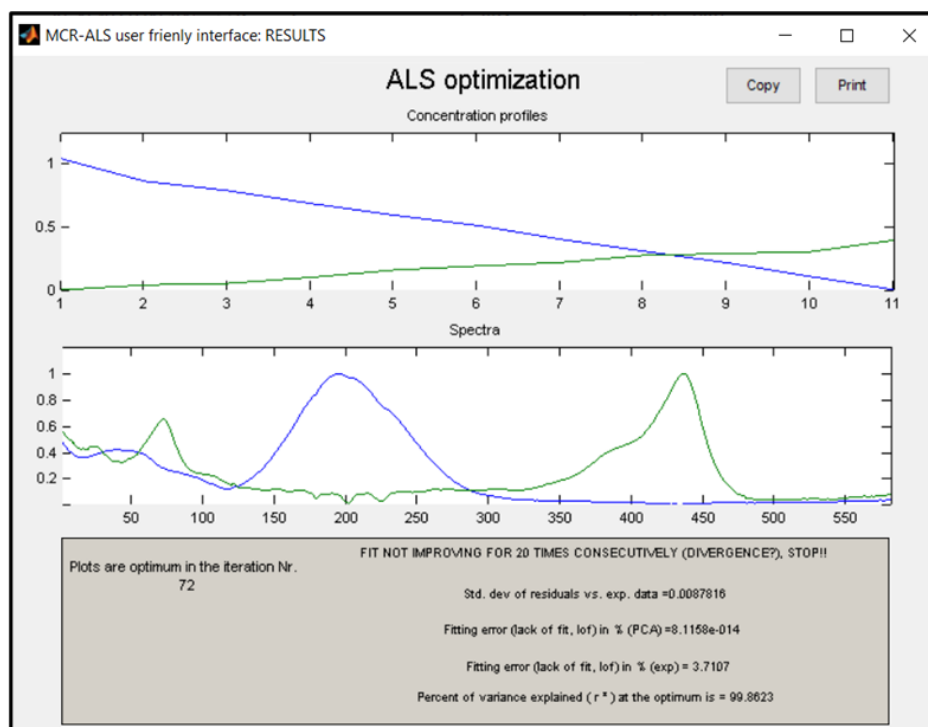


Figura 5S. Resultados do MCR-ALS

A convergência do modelo foi alcançada em 72 iterações, com baixo erro residual e ajuste adequado, como se observa nos resultados da Figura 5S, que mostram dois perfis de concentração na parte superior e dois espectros recuperados na parte inferior. Observando os espectros puros da Figura 1S, verifica-se que a linha azul corresponde ao AM e a verde ao AZ. Esses perfis são salvos como “copt” e “sopt”, respectivamente, a partir dos quais pode-se obter os espectros recuperados em função do comprimento de onda, por meio dos comandos: `>> plot(nm,copt(1,:), 'y'); >> hold on; plot(nm,copt(2,:), 'b');` e os perfis de concentração, por meio dos comandos: `>> plot(sopt (1,:), 'oy'); >> hold on; plot(sopt (2,:), 'ob');`. A partir dos comandos, os dados referentes ao AM são obtidos na coloração amarela (símbolo y nos comandos) e os referentes ao AZ, na cor azul (símbolo b nos comandos).

Para a obtenção da curva analítica pseudo-univariada, utiliza-se primeiro o comando `>> copt` e os resultados são apresentados na Figura 6S.

```
>> copt

copt =

1.0317    0
0.8548    0.0394
0.7810    0.0517
0.6792    0.0991
0.5872    0.1568
0.5063    0.1887
0.3968    0.2161
0.3033    0.2728
0.2118    0.2859
0.1043    0.3018
0.0006    0.3925
```

Figura 6S. Resultados de concentrações relativas obtidos por MCR-ALS para AM (primeira coluna) e AZ (segunda coluna)

A concentração de referência, ou “real”, do AZ é conhecida (primeira coluna da Tabela 1). O próximo passo é criar um vetor com a concentração real para as amostras 1, 3, 5, 7, 9 e 11 através do comando: `>> CrealAZ=[0 20 40 60 80 100];`. Por outro lado, cria-se também um vetor para a concentração relativa de AZ (destacados em azul na Figura 6), para as mesmas amostras, empregando o comando: `>> CRAZ=[0 0.0517 0.1568 0.2161 0.2859 0.3925];`. Através desses vetores obtém-se a curva de calibração pseudo-univariada (Figura 7S) com o comando: `>> plot(CrealAZ,CRAZ,'o');`.

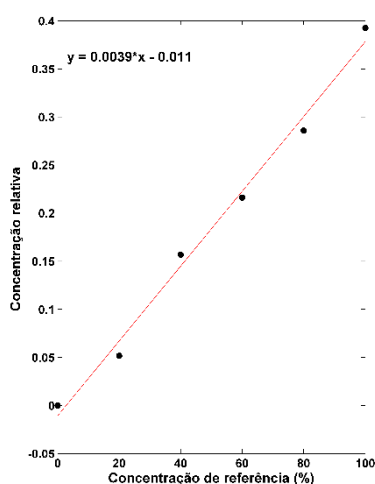


Figura 7S. Curva de calibração pseudo-univariada

O coeficiente de correlação para a curva analítica de calibração pseudo-univariada foi de 0,9956 (`>> corrcoef(CrealA,CRA);`) evidenciando correlação linear entre os dados.

Os mesmos cálculos e discussões acerca dos parâmetros de mérito também podem ser aplicados aos resultados obtidos no Matlab.

A ambiguidade dos perfis recuperados foi avaliada comparando o cosseno do ângulo dos vetores (calculado a partir do produto escalar normalizado) do perfil recuperado (a) e do sinal puro de cada analito (b), utilizando o comando $\gg \cos_theta = \text{dot}(a, b) / (\text{norm}(a) * \text{norm}(b));$.

Para o alaranjado de metila, o cosseno foi de 0,9994, enquanto para o azul de metileno foi de 0,9957, o que evidencia elevada concordância entre o espectro experimental de cada corante e o respectivo espectro recuperado por MCR-ALS, sugerindo que a ambiguidade rotacional foi significativamente reduzida pelas restrições aplicadas.

