# Opticna rotacija raztopine saharoze

Matija Zanjkovic, Mesarec Tilen, Petauer Maja

Junij 2025

## Uvod

glukoza : fruktoza = 36 : 41

```
def drude(lambda_nm, k, A):
return (k * lambda_nm**2) / (lambda_nm**2 - A**2)


# Prileganje z šupotevanjem napak v y (x napake ne vplivajo na fit)
params, cov = curve_fit(drude, lambda_vals, alpha_vals, sigma=
   alpha_errs, absolute_sigma=True, p0=(1e4, 200))
k_fit, A_fit = params
k_err, A_err = np.sqrt(np.diag(cov))

# Izpis rezultatov
print("Ujemajoci parametri:")
print(f"  k = {k_fit:.2f} ± {k_err:.2f}")
print(f"  A = {A_fit:.2f} ± {A_err:.2f} nm")

# Priprava za prikaz prileganja
lambda_fit = np.linspace(450, 700, 300)
alpha_fit = drude(lambda_fit, k_fit, A_fit)

# Risanje grafa z error bar-i v obeh smereh
plt.figure(figsize=(8, 5))

plt.errorbar(
    lambda_vals,
    alpha_vals,
    xerr=lambda_errs,
    yerr=alpha_errs,
    fmt='o',
    markersize=6,
    markerfacecolor='steelblue',
    markeredgecolor='black',
    ecolor='gray',
    elinewidth=1,
    capsize=4,
    label='Izmerjeni podatki z napako'
)

# Prileganje funkcije
plt.plot(lambda_fit, alpha_fit, color='cornflowerblue', linewidth=2.2,
   label='Drudejevo prileganje')

# Oznake in estetika
plt.xlabel('Valovna ždolina \lambda (nm)')
plt.ylabel('Specificna rotacija [\alpha](\lambda)')
plt.title('Prileganje podatkov Drudejevi enacbi z napakami v x in y')
plt.grid(True, linestyle='--', alpha=0.6)
plt.legend()
plt.tight_layout()
plt.show()
```

```python
from sympy import symbols, diff, simplify

# Define symbols
k, A, lam = symbols('k A lam')

# Define the function
alpha = k * lam**2 / (lam**2 - A**2)

# Partial derivatives
d_alpha_dk = simplify(diff(alpha, k))
d_alpha_dA = simplify(diff(alpha, A))
d_alpha_dlam = simplify(diff(alpha, lam))

print("\parcdelta[\alpha]/\parcdeltak =", d_alpha_dk)
print("\parcdelta[\alpha]/\parcdeltaA =", d_alpha_dA)
print("\parcdelta[\alpha]/\parcdelta\lambda =", d_alpha_dlam)
```

```python
from sympy import lambdify

# Izracun specificne rotacije in njene napake za modri laser (npr. \
    lambda = 450 nm)
lambda_blue = 450  # nm

# Uporabi Drudejev model in napake parametrov
alpha_blue = (k_fit * lambda_blue**2) / (lambda_blue**2 - A_fit**2)

# Delni odvodi po k in A že definirani: d_alpha_dk, d_alpha_dA)

d_alpha_dk_func = lambdify((k, A, lam), d_alpha_dk)
d_alpha_dA_func = lambdify((k, A, lam), d_alpha_dA)

d_alpha_dk_val = d_alpha_dk_func(k_fit, A_fit, lambda_blue)
d_alpha_dA_val = d_alpha_dA_func(k_fit, A_fit, lambda_blue)

# Skupna napaka (priblizek, zanemarimo korelacijo)
delta_alpha_blue = abs(d_alpha_dk_val) * k_err + abs(d_alpha_dA_val) *
    A_err

print(f"[\alpha](450 nm) = {alpha_blue:.2f} ± {delta_alpha_blue:.2f}")
```