Isotope Tracking - Species based

To obtain the masses of heavy M^h and light isotopes M^l, the isotope signature $\delta^{13}C$ is used where:

$$\delta^{13}C[\check{\mathrm{G}}] = \left(\frac{R_{smp} - R_{std}}{R_{std}}\right) \cdot 1000$$

$$R_{smp} = \frac{M^h}{M^l}$$

$$R_{std} = 11237.2 \cdot 10^{-6}$$

and where the total mass $M_{tot} = M^h + M^l$, we obtain:

$$R_{std}\left(\delta^{13}C/1000 + 1\right) = \frac{M_{tot} - M^l}{M^l}$$

$$M^l = \frac{M_{tot}}{1 + R_{std}(\delta^{13}C/1000 + 1)}$$

$$M^h = M_{tot} - M^l$$

Volatilization

```
def getVolatileMass(model, app_days, temp_air, theta_sat,
                    rel_diff_model="option-1", sorption_model="linear",
                    gas=True, isotopes=True):
    # Volatilize only during peak volatilization time i.e., first 24 hrs, @Prueger2005.
    if model.jd_cum in app_days:
        theta_layer = model.theta_z0
        tot_mass = model.pestmass_z0  # ug
        light_mass = model.lightmass_z0  # ug
        heavy_mass = model.heavymass_z0  # ug
        # Convert to m (needed for final mass computation on cell basis)
        depth_m = model.z0 * 1 / 10 ** 3
        # Air boundary layer, assumed equivalent to top soil thickness
        thickness_a = depth_m  # m
        # Diffusion coefficient in air (cm^2/s); https://www.gsi-net.com
        #  D_ar (metolachlor) = 0.03609052694,  at reference Temp., in Kelvin, D_a,r)
        diff_ar = 0.03609 * 86400 * 1 / 10 ** 4  # m2/d
        # Diffusion coefficient adjusted to air Temp., D_a
        diff_a = (temp_air / 293.15) ** 1.75 * diff_ar  # m2/d

        if rel_diff_model == "option-1":
            # Millington and Quirk, 1960 (in Leistra, 2001, p.48)
            # a,b parameters: Jin and Jury, 1996 (in Leistra, 2001)
            diff_relative_gas = (diff_a * (theta_sat - theta_layer) ** 2 /
                                 theta_sat ** (2 / 3))  # m2/d
```

```
        elif rel_diff_model == "option-2":
            # Currie 1960 (in Leistra, 2001)
            # a,b parameters: Baker, 1987 (in Leistra, 2001)
            diff_relative_gas = diff_a * 2.5 * (theta_sat - theta_layer) ** 3  # m2/d
        else:
            print("No appropriate relative diffusion parameter chosen")
            diff_relative_gas = diff_a  # m2/d
        # Transport resistance through air (r_a) and soil (r_s) layer
        r_a = thickness_a / diff_a  # d/m
        r_s = (0.5 * depth_m) / diff_relative_gas  # d/m

        # Retardation factor
        if sorption_model == "linear":
            retard_layer = 1 + (model.p_b * model.k_d) / theta_layer
        else:
            retard_layer = 1
        if gas:
            # Leistra et al., 2001
            theta_gas = theta_sat - theta_layer
            if isotopes:
                conc_light_aq = light_mass / ((cellarea() * model.z0) *  # m2 * mm = L
                                              (theta_gas * model.k_h +
                                               theta_layer * retard_layer))  # ug/L
                conc_heavy_aq = heavy_mass / ((cellarea() * model.z0) *
                                              (theta_gas * model.k_h +
                                               theta_layer * retard_layer))  # ug/L
            else:
                conc_layer_aq = tot_mass / ((cellarea() * model.z0) *
                                            (theta_gas * model.k_h +
                                             theta_layer * retard_layer))  # ug/L
        else:  # No gas phase
            if isotopes:
                conc_light_aq = light_mass / ((cellarea() * model.z0) *
                                              (theta_layer * retard_layer))  # ug/L
                conc_heavy_aq = heavy_mass / ((cellarea() * model.z0) *
                                              (theta_layer * retard_layer))  # ug/L
            else:  # No gas phase, no isotopes considered
                # Whelan, 1987
                conc_layer_aq = ((tot_mass / cellarea()) /
                                 (theta_layer * retard_layer * model.z0))  # ug/L

        # Convert ug/L to ug/m3, as will be multiplying by cell's area in m2
        if isotopes:
            conc_light_aq *= 10 ** 3  # ug/m3
            conc_heavy_aq *= 10 ** 3  # ug/m3
            conc_gas_light = conc_light_aq / model.k_h  # Henry's  ug/m3
            conc_gas_heavy = conc_heavy_aq / model.k_h  # Henry's  ug/m3
            volat_flux_light = (conc_gas_light / (r_a + r_s)) * cellarea()  # ug/day
            volat_flux_heavy = (conc_gas_heavy / (r_a + r_s)) * cellarea()  # ug/day
            volat_flux = volat_flux_light + volat_flux_heavy
        else:
            conc_layer_aq *= 10 ** 3  # ug/m3
            conc_gas_layer = conc_layer_aq / model.k_h  # Henry's  ug/m3
```

```
            volat_flux = (conc_gas_layer / (r_a + r_s)) * cellarea()  # ug/day
            volat_flux_light = None
            volat_flux_heavy = None
    else:
        volat_flux = scalar(0)
        volat_flux_light = scalar(0)
        volat_flux_heavy = scalar(0)

    return {"mass_volat": volat_flux,
            "light_volat": volat_flux_light,
            "heavy_volat": volat_flux_heavy}  # ug/dt
```