JVAL and JVPP User Manual

The photolysis module JVAL-14
The JVAL PreProcessor (JVPP)

Rolf Sander et al.*

Air Chemistry Department
Max-Planck Institute of Chemistry
PO Box 3060, 55020 Mainz, Germany
rolf.sander@mpic.de

Contents

*This manual is part of the electronic supplement of our article "The photolysis module JVAL-14, compatible with the MESSy standard, and the JVal PreProcessor (JVPP)" by R. Sander, P. Jöckel, O. Kirner, A. T. Kunert, J. Landgraf, and A. Pozzer in Geosci. Model Dev. (2014), available at: `http://www.geosci-model-dev.net`

Date: 2014-03-17

1 Introduction

Landgraf and Crutzen (1998) presented an efficient method for online calculations of photolysis and heating rates which is based on parameterizations in 8 wavelength bands. This method is implemented in the JVAL code of the MESSy system by Jöckel et al. (2010). Several calculations are necessary to obtain the JVAL parameters based on UV/VIS spectra, quantum yields, and their temperature and pressure dependencies. The JVal PreProcessor (JVPP) was written to simplify this process. This manual describes how to use JVPP and how to extend it for additional photolysis reactions.

2 JVAL

2.1 Compiling and running the JVAL column model with the shell script `xjval`

First, go to the JVAL base directory:

```
cd jval
```

Check that all settings in `Makefile` are correct for the Fortran90 compiler on your system. If available, define the path of your netcdf library in `NETCDF_INCLUDE` and `NETCDF_LIB`. If a netcdf library is not available, set "`OUTPUT = ASCII`".

Next, use the tcsh script `xjval` for compilation and execution of the JVAL column model. Type:

```
./xjval
```

The script will ask you if you want to compile the Fortran90 files and then if you want to run the model. After successful completion, the output will be in `jval.nc` (or `jval.dat` if `ASCII` output was chosen).

2.2 The internal structure of JVAL

The JVAL code consists of the Fortran90 files listed in Tab. 2. The call tree is shown in Fig. 1.

Note that for historical reasons, there are two variables for ozone in the code: `relo3_2d` and `v3_2d`. Only `v3_2d` is used to calculate J-values. The variable `relo3_2d` is used to calculate heating rates (not discussed here).

2.3 Namelist control

The file `jval.nml` contains the coupling namelist `&CPL` and the control namelist `&CTRL`. The coupling namelist is only needed when JVAL is connected to a 3-dimensional base model via the MESSy interface (Jöckel et al., 2010) but not for the column model. The variables in `&CPL` are:

Table 1: Subdivision of the spectral range into 8 bands. λ_{ini} and λ_{fin} are the initial and final wavelength. λ_i is a fixed wavelength inside each interval. Note that, for historical reasons, the bands are numbered from 0 to 7 in the JVAL code but 1 to 8 everywhere else.

number: name	$\frac{\lambda_{\text{ini}}}{\text{nm}}$	$\frac{\lambda_{\text{fin}}}{\text{nm}}$	$\frac{\lambda_i}{\text{nm}}$
1: Schumann-Runge	178.555	202.030	
2: Herzberg	202.030	240.970	205.1
3: Hartley	240.970	289.870	287.9
4:	289.870	305.500	302.0
5: UV-B	305.500	313.500	309.0
6:	313.500	337.500	320.0
7: UV-A	337.500	422.500	370.0
8: Chappuis	422.500	682.500	580.0

`l_skip_lg`: JVAL has, in the Submodel Interface Layer (SMIL), an extension for tracers in Lagrangian representation (for the definition of representation, see Jöckel et al. (2010)). If a tracer set (Jöckel et al., 2008) in Lagrangian representation is present, the J-values calculated in grid-point representaion are transformed to the Lagrangian representation for chemistry calculations along the trajectories. This can be switched off with `l_skip_lg = T`. The default is `F`, i.e., do not skip this calculation. This switch is only meaningful if a Lagrangian subsystem is running.

`l_force`: By default (`l_force = F`), only those J-values are calculated for which tracers in the selected chemical setup exist. This is done to avoid wasting CPU time. With this switch set to `T`, the calculation of all J-values is forced. This is useful to compute all J-values diagnostically without any chemistry calculations.

`l_heating`: This switch (default: `F`), if set to `T`, enables the calculation of the UV heating rates by oxygen and ozone as additional diagnostic quantitites.

`jval_O3`: This variable contains two strings (channel name and object name, see Jöckel et al. (2010)) which define the ozone input field. Examples are:

- `jval_O3 = 'tracer_gp', 'O3'`
  for an on-line coupling to the ozone tracer '`O3`', provided by the submodel TRACER in its channel '`tracer_gp`', or
- `jval_O3 = 'import_grid', 'O3clim'`
  for using an off-line prescribed ozone climatology '`O3clim`' provided by the submodel IMPORT_GRID in its channel '`import_grid`'.

`jval_cossza`: This defines the channel object providing the cosine of the solar zenith angle. For ex-

Table 2: List of JVAL files

Fortran90 code	
`jval.f90`	main column model file
`messy_cmn_photol_mem.f90`	common definitions shared by different photolysis codes
`messy_jval.f90`	static JVAL core file
`messy_jval_jvpp.inc`	JVPP-generated file, included by `messy_jval.f90`
`messy_main_blather.f90`	message output utilities (generic MESSy submodel)
`messy_main_constants_mem.f90`	physical constants (generic MESSy submodel)
`messy_main_tools.f90`	miscellaneous tools (generic MESSy submodel)
`mo_netcdf.f90`	input/output
`mo_netcdf.f90-ascii`	ASCII input/output
`mo_netcdf.f90-real`	netCDF input/output
Other files and directories	
`jval.nc`	output of JVAL
`jval.nml`	namelist file
`jvpp`	directory with JVPP code
`manual`	directory with JVAL manual
`xjval`	tcsh script to execute JVAL

```
PROGRAM jval
├─ jval_initialize ..... intialization
│  ├─ jval_read_nml_ctrl ..... read &CTRL namelist
│  └─ aerosol_data ..... initialise aerosol optical data (Shettle and Fenn, 1979)
├─ jval_init_memory ..... allocate memory for arrays
├─ jval_physc ..... main calculation
│  ├─ jval_solar_time_control ..... set solar cycle parameters
│  └─ jvalues ..... calculate J-values
│     ├─ column_cal ..... calculate O2 and O3 column densities
│     ├─ flux_cal ..... calculate actinic fluxes
│     │  ├─ slingo ..... cloud particle radiative properties (Slingo, 1989)
│     │  ├─ aero_2d ..... apply humidity to aerosol data
│     │  └─ pifm ..... practical improved flux method
│     │     └─ pifmini ..... pifm initialization
│     ├─ jval_cal_uv ..... calculate UV radiation
│     └─ jval_cal ..... calculate all J-values in inc file
│        └─ jval_cal_* ..... individual subroutines for all species
├─ jval_result ..... print results
└─ jval_free_memory ..... deallocate memory of arrays
```

Figure 1: Call tree of the JVAL column model.

ample,
`jval_cossza = 'orbit', 'cossza'`
selects the required values from object '`cossza`' provided by the submodel ORBIT via its channel '`orbit`'.

`jval_cdisse`: This defines the channel object providing the distance Sun-Earth (in AU), e.g.,
`jval_cdisse = 'orbit', 'cdisse'`
selects the required field from object '`cdisse`' provided by the submodel ORBIT via its channel '`orbit`'.

`jval_solar`: This defines the channel object providing information on the solar activity (solar cycle). If it is undefined (commented out in the namelist), the constant value `r_sol` in `&CTRL` is used instead (see below). For example,
`jval_solar = 'import_ts','solact'`
selects the object '`solact`' from the channel '`import_ts`' of the submodel `IMPORT_TS`. The channel object can contain one of the following:

- The 10.7 cm solar radio flux in solar flux units (sfu, 10^{-22} W m^{-2} Hz^{-1}) adjusted to the Sun-Earth distance of 1 AU: In this case a measure for the solar conditions between solar minimum (0, corresponding to about 70 sfu (Tapping, 2013)) and solar maximum (1, corresponding to 270 sfu) is linearly interpolated for the actual 10.7cm flux and applied as solar cycle modulation of the J-value calculations.

- 16 parameters (all adjusted to a Sun-Earth distance of 1 AU):
  - **1:** Lyman-α total flux at the top of the atmosphere (`phi_la`), e.g., 3×10^{11} for a low solar activity (Chabrillat and Kockarts, 1997).
  - **2:** Integrated flux over the Schumann-Runge bands (interval 1) at the top of the atmosphere (`SR_toa_flux`).
  - **3-9:** Flux at the top of the atmosphere at the fixed wavelengths λ_i inside intervals 2-8 taken from Table 1 (`flux`).
  - **10-16:** For each of the intervals, the ratio of the integrated flux (from $\lambda_{\rm ini}$ to $\lambda_{\rm fin}$, see Table 1) to the flux at the fixed wavelength λ_i (`f0`).

  The unit of `phi_la`, `SR_toa_flux`, and `flux` must be photons $\rm cm^{-2}\,s^{-1}$.

The variables in `&CTRL` are:

`r_sol`: For the solar cycle, a value of 0 defines the solar minimum, and a value of 1 the solar maximum. This variable is overwritten if `jval_solar` in `&CPL` is set.

`qy_ch3coch3`: There are three options to calculate the quantum yield for acetone (CH_3COCH_3) photolysis, based on different publications. For details, see `jvpp/dat_lit/hardcoded/jval_cal_CH3COCH3.f90`.

Normally, there is no need to change the default settings of the namelists.

3 JVPP

3.1 Compiling and running JVPP with the shell script `xjvpp`

First, go to the base directory of the JVPP code:

```
cd jval/jvpp
```

Note that all path names given in this section are relative to this base directory. If you have the full MESSy code, the JVPP base directory is located inside `messy/tools/`. Check that all settings in `Makefile` are correct for the Fortran90 compiler on your system. Next, the tcsh script `xjvpp` will guide you through the process of running the code, as illustrated in Fig. 2. To execute the script, type:

```
./xjvpp
```

`xjvpp` will ask several questions, and recommended answers are given below. If you only press the Return key, you select the default.

```
Compile f90 files?
[y|n|q, default=y]
```

Choose "`y`" to compile the Fortran90 files and create the executable `jvpp.exe`.

```
Choose input directory,
default is same as last time
[m17|lit|default=lit]
```

Choose "`lit`" to select the input directory "`dat_lit/`" which contains the most recent UV/VIS data from the literature (the option "`m17`" refers to input data that was used by MESSy-1.7 and is only used for testing purposes).

```
Clean-up workdir and run jvpp.exe?
[y|n|q, default=y]
```

Unless there were any errors, choose "`y`" now to remove temporary files from the working directory and to start the JVPP executable. After `jvpp.exe` has finished, the screen output can also be found in `jvpp.log`. More detailed information is available from `jvpp_detail.log`. The main output file is `messy_jval_jvpp.inc`, which contains the Fortran90 code with the subroutines `jval_cal_*` for all photolysis reactions. The JVAL column model will use the new include file directly. To connect it to another base model, simply copy or link `messy_jval_jvpp.inc` into the directory where `messy_jval.f90` is, e.g. `messy/smcl/` or `caaba/`.

3.2 The internal structure of JVPP

The JVPP code consists of the files listed in Tab. 3. The main program `jvpp.f90` works in three steps as described below and shown in Fig. 3.

3.2.1 Step 1

The file `jvpp_step1.f90` contains the code to perform the first step. Here, the subroutine `conv_sig_176` converts (interpolates) cross sections σ from literature data to the 176 fixed wavelengths shown in Table 4. If available, quantum yields φ and temperature dependencies of the cross sections are also interpolated.

The code loops over all photolysis reactions (from 1 to `IP_MAX`). For each reaction, it is checked if there are any input files for the spectra (cross sections), their quantum yields, and/or their temperature dependencies. Files with the following extensions can be used as model input:

`*.sig`: temperature-independent spectrum

`*.s2t`: 2 spectra at 2 temperatures

`*.s3t`: 3 spectra at 3 temperatures

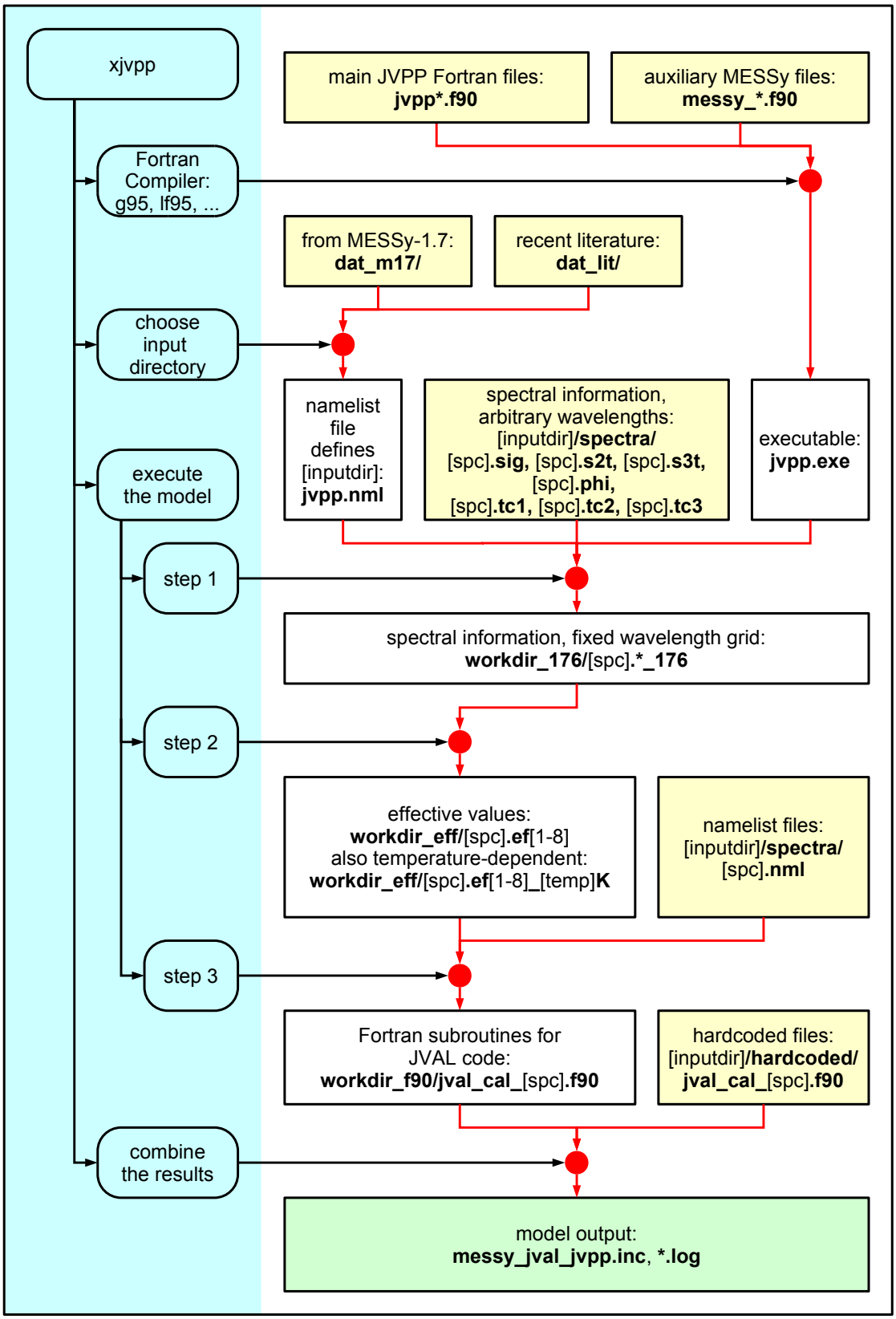

Figure 2: Illustration of the tasks performed by `xjvpp`. `xjvpp` and all scripts called by `xjvpp` are shown on a blue background. User-generated (static) input files are shown on a yellow background whereas automatically generated temporary files are shown on a white background.

Table 3: List of JVPP files

Fortran90 code	
`jvpp.f90`	main program
`jvpp_step1.f90`	step 1 (see text)
`jvpp_step2.f90`	step 2 (see text)
`jvpp_step3.f90`	step 3 (see text)
`jvpp_mem.f90`	common variable declarations
`messy_cmn_photol_mem.f90`	common definitions shared by different photolysis codes
`messy_main_constants_mem.f90`	physical constants
`messy_main_math_lsq.f90`	least-square fit methods
`messy_main_math_spline.f90`	spline methods
Input	
`cheb_coeff.txt`	Chebyshev coefficients (for O_2 cross sections)
`jvpp.nml`	automatically created namelist file
`dat_m17/`	obsolete data from MESSy-1.7
`dat_lit/`	data from recent literature
`dat_lit/spectra/*.nml`	individual namelist files
`dat_lit/spectra/*.sig`	original spectra, as from the literature
`dat_lit/spectra/*.s2t`	spectra at 2 temperatures
`dat_lit/spectra/*.s3t`	spectra at 3 temperatures
`dat_lit/spectra/*.tc1`	temperature coefficients (linear)
`dat_lit/spectra/*.tc2`	temperature coefficients (quadratic)
`dat_lit/spectra/*.phi`	quantum yields
`dat_lit/hardcoded/*.f90`	hardcoded subroutines for JVAL
Output	
`workdir_176/*`	temporary data created in step1
`workdir_eff/*`	temporary data created in step2
`workdir_f90/*`	temporary f90 subroutines for JVAL
`workdir_jnl/*`	`*.jnl` files for plotting the spectra and JVAL coefficients for the 8 bands
`dat_lit/old/`	backup from last run
`references_jvpp.tex`	sources for the spectra
Other files	
`xjvpp`	tcsh script to execute JVPP
`jnl/*`	ferret plotting files
`util/`	some utilities (only needed for code development)

`*.phi`: quantum yields (between 0 and 1)

`*.tc1`: linear temperature coefficients (1st order polynomial)

`*.tc2`: quadratic temperature coefficients (2nd order polynomial)

`*.tc3`: cubic temperature coefficients (3rd order polynomial)

The input files start with an arbitrary number of header lines which must begin with the character "`#`". The following lines must contain data. The first column defines the wavelength in nm. The content of the following columns depends on the file type. For the `*.sig` files, the second column contains (temperature-independent) cross sections in cm^2. The temperature-dependent files contain additional lines and columns.

The data from each of these files is interpolated to the 176 fixed wavelengths. The default interpolation method is integration (`spline_method = integration`). It performs a linear interpolation between the points of the original spectrum and conserves the integrated value. As an alternative, it is possible to select other methods, based on code from John Burkardt (`http://people.scs.fsu.edu/~burkardt/f_src/spline/spline.html`): A linear spline method (`linear_val`), a cubic B spline (`b_val`), a piecewise constant spline (`constant_val`), and a piecewise quadratic spline method (`quadratic_val`) are available.

The results are written to temporary files with the suffix "`.*_176`" in the directory `workdir_176/`.

3.2.2 Step 2

The file `jvpp_step2.f90` contains the code to perform the second step. Here, the subroutine `conv_176_eff` reads data for the 176 bins from `workdir_176/` and

```
PROGRAM jvpp
├─ conv_sig_176 ..... step 1: interpolate to 176 intervals
│  └─ process_species ..... one call per species
│     └─ spline_* ..... integration or spline
├─ conv_176_eff ..... step 2: calculate effective values
│  ├─ initialize ..... initialization
│  │  ├─ read_tc ..... read temperature coefficients
│  │  └─ read_phi ..... read quantum yields φ
│  └─ calc_interval ..... loop over bins and temperatures
│     ├─ calc_tdep ..... calculate temperature-dependent cross sections
│     │  └─ sr_o2_km ..... parameterization for Schumann-Runge
│     ├─ calc_phi ..... calculate quantum yields
│     ├─ sig_eff ..... calculate effective values
│     │  └─ write_xxx ..... write effective values
└─ conv_eff_poly ..... step 3: polynomial fits
   ├─ read_nml ..... read namelist for each species
   └─ process_species ..... process each species
      ├─ process_interval_* ..... process each interval (constant temperature)
      │  ├─ read_file_*d ..... read effective values
      │  └─ poly_fit ..... polynomial fit
      ├─ process_tdep_interval_* ..... process each interval (temperature dependent)
      │  ├─ read_file_*d ..... read effective values
      │  └─ poly_fit ..... polynomial fit
      └─ write_* ..... write Fortran90 include file for JVAL
```

Figure 3: Main subroutines in the call tree of JVPP.

then converts them to effective values for the eight bands shown in Table 1. First, output from step 1 is read in:

- If there are any temperature-dependent (`*.s2t` or `*.s3t`) or temperature-independent (`*.sig`) cross-section input files, the data is read into the variables `cs_xxx_tdep` or `cs_xxx`, respectively.
- If there are any files containing temperature coefficients (`*.tc1`, `*.tc2`, or `*.tc3`), the data is read into the variable `tc_xxx`.
- If there are any files containing quantum yields (`*.phi`), the data is read into the variable `phi_xxx`.

Next, the code loops over all bands (from 1 to 8), first for calculations at a fixed reference temperature (240 K for interval 1 and 250 K for intervals 2-8), then for calculations at several temperatures (from 180 K to 320 K). For each band:

- Temperature-dependent cross sections `cst_xxx` are calculated in subroutine `calc_tdep`:
  - via `cs_xxx_tdep` from `*.s2t` or `*.s3t` files
  - via temperature coefficients `tc_xxx` from `*.tc1`, `*.tc2`, or `*.tc3` files
  - via individual wavelength-dependent functions defined in the code
- Quantum yields are calculated in subroutine `calc_phi`.
- Effective values are calculated in subroutine `sig_eff`:
  - Ozone columns and optical depths `tau_o3` are defined.
  - Oxygen columns and optical depths `tau_o2` are defined.
  - The code loops over all O_3 and O_2 columns and prints intermediate values for all photolysis reactions to a temporary file.

The resulting effective values are written to temporary files in the directory `workdir_eff/`. The suffix of these files is "`.ef`[*1-8*]" for temperature-independent data and "`.ef`[*1-8*]`_`[*temp*]`K`" for temperature-dependent data. Here [*1-8*] denotes the wavelength band between 1 and 8 and [*temp*] is the temperature.

3.2.3 Step 3

The file `jvpp_step3.f90` contains the code to perform the third step. First, the subroutine `conv_eff_poly` looks for namelist files. For all species with a namelist, the subroutine `process_species` is called. It reads the effective values from `workdir_eff/` and then finds polynomial fits for them. From the parameterization, the Fortran90 code of `SUBROUTINE jval_cal_XYZ` is generated, where `XYZ` is the name of the species.

Table 4: Wavelengths (in nm) of the fixed grid created in step 1 of JVPP. The full grid contains 176 wavelengths. However, here only the first 142 wavelengths are shown here because those above 680 nm are currently not used.

1) Schumann-Runge		**4)**		**8) Chappuis**	
1	179.37	44	291.97	91	425.00
2	181.00	45	296.30	92	430.00
3	182.65	46	299.00	93	435.00
4	184.33	47	300.00	94	440.00
5	186.05	48	301.00	95	445.00
6	187.79	49	302.00	96	450.00
7	189.57	50	303.00	97	455.00
8	191.39	51	304.00	98	460.00
9	193.24	52	305.00	99	465.00
10	195.12	**5) UV-B**		100	470.00
11	197.04	53	306.00	101	475.00
12	199.00	54	307.00	102	480.00
13	201.01	55	308.00	103	485.00
2) Herzberg		56	309.00	104	490.00
14	203.05	57	310.00	105	495.00
15	205.13	58	311.00	106	500.00
16	207.25	59	312.00	107	505.00
17	209.42	60	313.00	108	510.00
18	211.64	**6)**		109	515.00
19	213.90	61	314.00	110	520.00
20	216.22	62	315.00	111	525.00
21	218.58	63	316.00	112	530.00
22	220.99	64	317.00	113	535.00
23	223.46	65	318.00	114	540.00
24	225.99	66	319.00	115	545.00
25	228.57	67	320.00	116	550.00
26	231.21	68	321.00	117	555.00
27	233.92	69	322.50	118	560.00
28	236.69	70	324.50	119	565.00
29	239.52	71	326.50	120	570.00
3) Hartley		72	330.00	121	575.00
30	242.42	73	335.00	122	580.00
31	245.40	**7) UV-A**		123	585.00
32	248.45	74	340.00	124	590.00
33	251.57	75	345.00	125	595.00
34	254.78	76	350.00	126	600.00
35	258.06	77	355.00	127	605.00
36	261.44	78	360.00	128	610.00
37	264.90	79	365.00	129	615.00
38	268.46	80	370.00	130	620.00
39	272.11	81	375.00	131	625.00
40	275.86	82	380.00	132	630.00
41	279.72	83	385.00	133	635.00
42	283.69	84	390.00	134	640.00
43	287.77	85	395.00	135	645.00
		86	400.00	136	650.00
		87	405.00	137	655.00
		88	410.00	138	660.00
		89	415.00	139	665.00
		90	420.00	140	670.00
				141	675.00
				142	680.00

Finally, the script `cat_jval.tcsh` (also created by `jvpp_step3.f90`) is used to concatenate all individual subroutines `jval_cal_*` into the include file `messy_jval_jvpp.inc`.

3.3 Namelist control

The main namelist file `jvpp.nml` is created automatically by `xjvpp` and should not be edited manually.

In contrast, the individual namelist files for each photolysis reaction can be changed manually. They are in the same input directory as for the spectra, e.g., `dat_lit/spectra/`. They contain the `JVPP` namelist which is used in step 3. The entries are:

- The degree of the fitting functions for temperature-independent and temperature-dependent effective values can be defined with `deg_tconst` and `deg_tdep`, respectively. Each of the 8 bands can have individual values. The default is:
  `deg_tconst` = (/ 1, 1, 1, 1, 3, 3, 3, 3 /)
  `deg_tdep` = (/ 2, 2, 2, 2, 2, 2, 2, 2 /)
- Eight correction factors for zenith angles above 87.5° are available, as shown in Fig. 4. For new species, a suitable factor should be chosen according to Table 1 of Lamago et al. (2003), based on the wavelength region in which the species absorbs. If the factor is not defined in the namelist, the default value `fj_corr` = 7 is used.
- If absorption at the Lyman-alpha wavelength (e.g., CO_2, O_2) or in the infrared region (e.g., HNO_4) is important, its contribution can be defined as `lya_ir`.
- If JVPP is unable to calculate the parameters for JVAL (e.g., because of density dependence correction for acetone), setting `l_hardcoded` to `.TRUE.` will simply use a manually written subroutine from `dat_lit/hardcoded/`. Examples can be seen in `CH3COCH3.nml`, `GLYOX.nml`, and `NO.nml`.

For inclusion into the MESSy system, some metadata should be provided:

- `meccanum` is the number of the photolysis reaction in the MECCA chemistry mechanism (Sander et al., 2011), e.g., `meccanum = "J3101"` for the photolysis of NO_2.
- `texrxn` contains the photolysis reaction in LaTeX syntax, e.g.: `"NO_2 \TOHV\ NO + O"`.
- `texnote` provides the reference for the spectrum as a BibTeX label. It may also contain additional information in LaTeX syntax, e.g.: `"Lyman-alpha from Fig. 1 of \cite{2354}"` for CH_4.

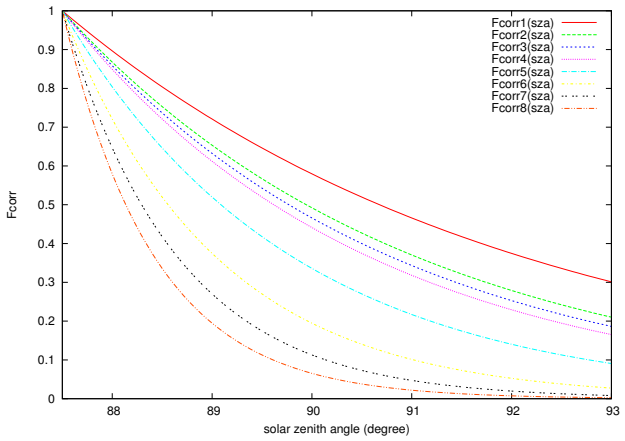

Figure 4: Correction factor for large solar zenith angles according to Lamago et al. (2003).

4 Modifying JVAL and JVPP

4.1 Adding a new photolysis reaction

To add the photolysis of a new species (called "XYZ" here) to the code, the following changes are necessary:

- Add the photolysis to `messy_cmn_photol_mem.f90`:
  - add `ip_XYZ`
  - increase `IP_MAX`
  - add string to `jname`
- Create a new namelist file `XYZ.nml` with appropriate values and save it in the `dat_lit/spectra/` directory.
- Get the UV/VIS spectrum (e.g., from Keller-Rudek et al. (2013)) and enter it to a new file `XYZ.sig` in the `dat_lit/spectra/` directory.
- If the cross sections are temperature-dependent, choose one of the following options:
  - Create a `XYZ.s2t` or `XYZ.s3t` file if the spectrum is known at two or three temperatures, respectively.
  - Create a `XYZ.tc1`, `XYZ.tc2`, or `XYZ.tc3` file if the spectrum can be described with a function containing 1, 2, or 3 parameters. Add the corresponding function to `SUBROUTINE calc_tdep` in `jvpp_step2.f90`.
  - For more complex cases, add an individual wavelength-dependent function at the end of `SUBROUTINE calc_tdep` in `jvpp_step2.f90`.
- If the quantum yield is not always equal to one, add a `XYZ.phi` file to the directory `dat_lit/spectra/`.

- Execute the JVPP code via `xjvpp` as described in Sect. 3.1.
- If the new photolysis reaction is used in the global ECHAM5/MESSy Atmospheric Chemistry (EMAC) model (Jöckel et al., 2010), it is necessary to activate its calculation in `messy_jval_si.f90` with:
```
IF (TRIM(basename) == 'XYZ') &
  lps(ip_XYZ,j) = .TRUE.
```

4.2 Evaluating changes in JVPP

To compare the spectra to those that were used in MESSy-1.7, go to the directory `workdir_jnl`, and run the (automatically generated) ferret script `jvpp_step1.jnl`. To check the quality of the polynomial fits, run `jvpp_step3_tconst.jnl` and `jvpp_step3_tdep.jnl` in the same directory. To check if the temperature dependence of a spectrum was calculated correctly in step 2, adjust and run `jnl/tdep.jnl`.

4.3 Evaluating changes in JVAL

To evaluate how your modifications change the calculated J-values, follow these steps:

- Run the JVAL column model via `xjval` as described above.
- Rename the resulting netcdf file, using a descriptive name, e.g.:
```
mv jval.nc jval_BASE.nc
```
- Optionally, make a backup of the include file, using a descriptive name, e.g.:
```
cp messy_jval_jvpp.inc
   messy_jval_jvpp_BASE.inc
```
- Modify the JVPP code as desired and create a new include file via `xjvpp`.
- Run the JVAL column model again, then rename the output using a different name, e.g., "`NEW`" instead of "`BASE`".
- Edit the ferret script `compare_jval.jnl`:
  - Select two netcdf files for the comparison, e.g.:
```
USE "jval_BASE.nc"
USE "jval_NEW.nc"
```
  - Select one or more photolysis reactions for the comparison, e.g.:
```
GO _compare_jval J_O3P
GO _compare_jval J_O1D
```
- Execute the ferret script:
```
ferret
go compare_jval.jnl
```

References

Chabrillat, S. and Kockarts, G.: Simple parameterization of the absorption of the solar Lyman-alpha line, Geophys. Res. Lett., 24, 2659–2662, 1997.

Jöckel, P., Kerkweg, A., Buchholz-Dietsch, J., Tost, H., Sander, R., and Pozzer, A.: Technical Note: Coupling of chemical processes with the Modular Earth Submodel System (MESSy) submodel TRACER, Atmos. Chem. Phys., 8, 1677–1687, `http://www.atmos-chem-phys.net/8/1677`, 2008.

Jöckel, P., Kerkweg, A., Pozzer, A., Sander, R., Tost, H., Riede, H., Baumgaertner, A., Gromov, S., and Kern, B.: Development cycle 2 of the Modular Earth Submodel System (MESSy2), Geosci. Model Dev., 3, 717–752, `http://www.geosci-model-dev.net/3/717`, 2010.

Keller-Rudek, H., Moortgat, G. K., Sander, R., and Sörensen, R.: The MPI-Mainz UV/VIS spectral atlas of gaseous molecules of atmospheric interest, Earth Syst. Sci. Data, 5, 365–373, `http://www.earth-syst-sci-data.net/5/365`, 2013.

Lamago, D., Dameris, M., Schnadt, C., Eyring, V., and Brühl, C.: Impact of large solar zenith angles on lower stratospheric dynamical and chemical processes in a coupled chemistry-climate model, Atmos. Chem. Phys., 3, 1981–1990, 2003.

Landgraf, J. and Crutzen, P. J.: An efficient method for online calculations of photolysis and heating rates, J. Atmos. Sci., 55, 863–878, 1998.

Sander, R., Baumgaertner, A., Gromov, S., Harder, H., Jöckel, P., Kerkweg, A., Kubistin, D., Regelin, E., Riede, H., Sandu, A., Taraborrelli, D., Tost, H., and Xie, Z.-Q.: The atmospheric chemistry box model CAABA/MECCA-3.0, Geosci. Model Dev., 4, 373–380, `http://www.geosci-model-dev.net/4/373`, 2011.

Shettle, E. P. and Fenn, R. W.: Models for the aerosols of the lower atmosphere and the effects of the humidity variations on their optical properties, Envronmental Research Papers, No. 676 AFGL-TR-79-0114, Air Force Geophysics Laboratory, Hanscom AFB, Massachusetts 01731, 1979.

Slingo, A.: A GCM parameterization for the shortwave radiative properties of water clouds, J. Atmos. Sci., 46, 1419–1427, 1989.

Tapping, K. F.: The 10.7 cm solar radio flux ($F_{10.7}$), Space Weather, 11, 394–406, 2013.