The Reactions of the German Stock Market to COVID-19 and Containment Policies: A Vector Autoregressive Analysis

12. Estimation of VAR-Model

In [1]:
```
# Importing the necessary python packages
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
import statsmodels.api as sm
from statsmodels.tsa.api import VAR

%matplotlib inline
```

In [2]:
```
# Load in the previosuly prepared datasets
sample_1 = pd.read_csv("transformed_data_sample_1.csv", parse_dates=["date"], index_col="date")
sample_2 = pd.read_csv("transformed_data_sample_2.csv", parse_dates=["date"], index_col="date")
sample_full = pd.read_csv("transformed_data_sample_full.csv", parse_dates=["date"], index_col="date")
```

In [3]:
```
# Create a vector auto regressive model (VAR) for all samples timeframes
# Include dummy variables as exogen variables [ Y(t) = ß0 + ß1*Y(t-1) + ... + ßp*Y(t-p) + B*D(t) + e(t) ] -> No lags!

# VAR model for all first wave sample
model_1 = VAR(endog=sample_1[["new_cases_pct", "stringency_diff", "hdax_pct"]],
              exog=sample_1[["Monday", "Tuesday", "Wednesday", "Thursday"]],
              missing='none')

# VAR model for all second wave sample
model_2 = VAR(endog=sample_2[["new_cases_pct", "stringency_diff","hdax_pct", ]],
              exog=sample_2[["Monday", "Tuesday", "Wednesday", "Thursday"]],
              missing='none')

# VAR model for full sample timeframe
model_full = VAR(endog=sample_full[["new_cases_pct", "stringency_diff", "hdax_pct", ]],
              exog=sample_full[["Monday", "Tuesday", "Wednesday", "Thursday"]],
              missing='none')
```

```
/Users/maximbuz/Anaconda/anaconda3/lib/python3.8/site-packages/statsmodels/tsa/base/tsa_model.py:581: ValueWarning: A da
te index has been provided, but it has no associated frequency information and so will be ignored when e.g. forecasting.
  warnings.warn('A date index has been provided, but it has no'
/Users/maximbuz/Anaconda/anaconda3/lib/python3.8/site-packages/statsmodels/tsa/base/tsa_model.py:581: ValueWarning: A da
te index has been provided, but it has no associated frequency information and so will be ignored when e.g. forecasting.
  warnings.warn('A date index has been provided, but it has no'
/Users/maximbuz/Anaconda/anaconda3/lib/python3.8/site-packages/statsmodels/tsa/base/tsa_model.py:581: ValueWarning: A da
te index has been provided, but it has no associated frequency information and so will be ignored when e.g. forecasting.
  warnings.warn('A date index has been provided, but it has no'
```

In [4]:

```
# Show how many lags different information criterions suggest for first sample
model_1.select_order(trend='c').summary()
```

Out[4]:

VAR Order Selection (* highlights the minimums)

	AIC	BIC	FPE	HQIC
0	-4.860	-4.281*	0.007770	-4.640*
1	-4.970	-4.044	0.007003*	-4.619
2	-4.772	-3.498	0.008661	-4.289
3	-4.657	-3.036	0.009968	-4.042
4	-4.458	-2.489	0.01267	-3.711
5	-4.232	-1.916	0.01690	-3.353
6	-4.757	-2.093	0.01091	-3.747
7	-4.866	-1.854	0.01107	-3.723
8	-4.584	-1.225	0.01739	-3.309
9	-4.469	-0.7627	0.02463	-3.063
10	-4.391	-0.3375	0.03682	-2.853
11	-4.989*	-0.5873	0.03214	-3.319

In [5]:

```
# Show how many lags different information criterions suggest for second sample
model_2.select_order(trend='c').summary()
```

Out[5]:

VAR Order Selection (* highlights the

minimums)

	AIC	BIC	FPE	HQIC
0	-9.393	-9.119	8.332e-05	-9.282
1	-9.611*	-9.174*	6.700e-05*	-9.434*
2	-9.569	-8.968	6.989e-05	-9.325
3	-9.513	-8.748	7.395e-05	-9.203
4	-9.467	-8.537	7.752e-05	-9.090
5	-9.527	-8.433	7.308e-05	-9.083
6	-9.583	-8.325	6.923e-05	-9.073
7	-9.547	-8.125	7.191e-05	-8.970
8	-9.548	-7.962	7.204e-05	-8.905
9	-9.470	-7.720	7.813e-05	-8.760
10	-9.451	-7.538	7.992e-05	-8.675
11	-9.504	-7.427	7.617e-05	-8.661
12	-9.457	-7.216	8.029e-05	-8.548
13	-9.443	-7.037	8.200e-05	-8.467
14	-9.462	-6.892	8.110e-05	-8.419

In [6]:

```
# Show how many lags different information criterions suggest for full sample timeframe
model_full.select_order(trend='c').summary()
```

Out[6]:

VAR Order Selection (* highlights the minimums)

	AIC	BIC	FPE	HQIC
0	-8.166	-7.989*	0.0002842	-8.095*
1	-8.188*	-7.904	0.0002780*	-8.075
2	-8.136	-7.746	0.0002929	-7.980
3	-8.109	-7.613	0.0003010	-7.911

4	-8.089	-7.487	0.0003071	-7.848
5	-8.137	-7.428	0.0002928	-7.854
6	-8.139	-7.324	0.0002922	-7.814
7	-8.141	-7.220	0.0002916	-7.773
8	-8.102	-7.076	0.0003033	-7.692
9	-8.071	-6.938	0.0003130	-7.619
10	-8.047	-6.808	0.0003208	-7.553
11	-8.102	-6.757	0.0003039	-7.565
12	-8.094	-6.643	0.0003066	-7.515
13	-8.079	-6.521	0.0003117	-7.457
14	-8.068	-6.404	0.0003155	-7.403
15	-8.022	-6.251	0.0003308	-7.315
16	-8.021	-6.144	0.0003316	-7.271

use both AIC and BIC. Most of the times they will agree on the preferred model, when they don't, just report it.

In [7]:

```
# Write a function to find leg-lenghts at which
# the portmonteau test's H0: "Absence of significant residual autocorrelations"
# is failed to be rejected, signifying a lag-structure suitable for VAR-estimations
# As per Lütkepohl a large nlags (h) is necessary to use the Portmanteau test, so we use 1/4 of the observations as h

def lags_whiteness(models_dict, significance = 0.05, maxlag=10):
    result_list = []
    for key, model in models_dict.items():
        n = 1
        no_acorr_n = []
        while True:
            results = model.fit(n)
            test = results.test_whiteness(int(results.nobs*0.25), signif=significance, adjusted=False)
            test_statistic = test.test_statistic
            critical_value = test.crit_value
            if test_statistic < critical_value:
                no_acorr_n.append(n)
            n = n + 1
            if n > maxlag:
```

```
                break
            result_list.append(key + ": No significant autocorrelation at " + str(no_acorr_n) + " lags.")
    return result_list
```

In [8]:

```
# For the three different samples, use the above function
models = {"Model_Sample_1": model_1, "Model_Sample_2": model_2, "Model_Sample_full": model_full}

lags_whiteness(models, maxlag=7)
```

Out[8]:

```
['Model_Sample_1: No significant autocorrelation at [1, 2, 3, 4, 5, 6] lags.',
 'Model_Sample_2: No significant autocorrelation at [1, 2, 3, 4, 5, 6, 7] lags.',
 'Model_Sample_full: No significant autocorrelation at [6] lags.']
```

Serial autocorrelation tests with BG LM-Tests

Sample 1

one lag:

BG Test with p=1: Chi-squared = 14.808, df = 9, p-value = 0.09634*

four lags:

BG Test with p=1: Chi-squared = 16.463, df = 9, p-value = 0.05781*

Sample 1_1

one lag:

BG test with p=1: Chi-squared = 12.124, df = 9, p-value = 0.2064*

BG test with p=2: Chi-squared = 22.867, df = 18, p-value = 0.1957*

BG test with p=4: Chi-squared = 48.221, df = 36, p-value = 0.08377*

BG test with p=5: Chi-squared = 54.134, df = 45, p-value = 0.1651*

BG test with p=6: Chi-squared = 70.768, df = 54, p-value = 0.0625*

BG test with p=7: Chi-squared = 82.193, df = 63, p-value = 0.05264*

three lags:

BG test with p=1: Chi-squared = 12.13, df = 9, p-value = 0.2061*

four lags:

BG test with p=1: Chi-squared = 13.64, df = 9, p-value = 0.1357*

BG test with p=2: Chi-squared = 21.025, df = 18, p-value = 0.2782*

Sample 2

One lag:

BG test with p=1: Chi-squared = 11.345, df = 9, p-value = 0.2528*

BG test with p=2: Chi-squared = 25.413, df = 18, p-value = 0.114*

BG test with p=3: Chi-squared = 28.56, df = 27, p-value = 0.3826*

BG test with p=4: Chi-squared = 47.073, df = 36, p-value = 0.1024*

Two lags:

BG test with p=1: Chi-squared = 6.1163, df = 9, p-value = 0.7282*

BG test with p=2: Chi-squared = 21.799, df = 18, p-value = 0.2411*

BG test with p=3: Chi-squared = 28.661, df = 27, p-value = 0.3775*

BG test with p=4: Chi-squared = 45.866, df = 36, p-value = 0.1255*

three lags:

BG test with p=1: Chi-squared = 11.511, df = 9, p-value = 0.2423*

BG test with p=2: Chi-squared = 25.897, df = 18, p-value = 0.1021*

BG test with p=3: Chi-squared = 35.379, df = 27, p-value = 0.1295*

BG test with p=4: Chi-squared = 47.866, df = 36, p-value = 0.08921*

four lags:

BG test with p=1: Chi-squared = 15.449, df = 9, p-value = 0.07933*

Sample full

no lag structure without autocorrelation found...

In [9]:
```
# Getting the result from the regression of the models
results_1 = model_1.fit(1)
results_2 = model_2.fit(1)
```

In [10]:
```
# Getting the summary of the results for the first sample period
results_1.summary()
```

Out[10]:
```
  Summary of Regression Results
==================================
Model:                         VAR
Method:                        OLS
Date:           Fri, 15, Oct, 2021
Time:                     09:12:42
--------------------------------------------------------------------
No. of Equations:         3.00000    BIC:                   -2.77951
Nobs:                     59.0000    HQIC:                  -3.29472
```

```
Log likelihood:          -120.226    FPE:                    0.0267941
AIC:                     -3.62461    Det(Omega_mle):         0.0182966
--------------------------------------------------------------------
Results for equation new_cases_pct
======================================================================================
                        coefficient       std. error           t-stat            prob
--------------------------------------------------------------------------------------
const                      0.846245         0.406040            2.084           0.037
Monday                    -1.130521         0.577211           -1.959           0.050
Tuesday                   -0.036829         0.578167           -0.064           0.949
Wednesday                 -0.146113         0.573091           -0.255           0.799
Thursday                  -0.655341         0.548140           -1.196           0.232
L1.new_cases_pct          -0.138118         0.129988           -1.063           0.288
L1.stringency_diff        -0.082413         0.055962           -1.473           0.141
L1.hdax_pct              -13.885643         5.630836           -2.466           0.014
======================================================================================

Results for equation stringency_diff
======================================================================================
                        coefficient       std. error           t-stat            prob
--------------------------------------------------------------------------------------
const                      0.173385         1.001908            0.173           0.863
Monday                     1.473667         1.424273            1.035           0.301
Tuesday                   -0.157866         1.426632           -0.111           0.912
Wednesday                 -0.348080         1.414107           -0.246           0.806
Thursday                  -0.352099         1.352539           -0.260           0.795
L1.new_cases_pct           0.557938         0.320746            1.740           0.082
L1.stringency_diff         0.197269         0.138088            1.429           0.153
L1.hdax_pct                9.824909        13.894138            0.707           0.479
======================================================================================

Results for equation hdax_pct
======================================================================================
                        coefficient       std. error           t-stat            prob
--------------------------------------------------------------------------------------
const                      0.001185         0.009951            0.119           0.905
Monday                    -0.000257         0.014146           -0.018           0.985
Tuesday                    0.005077         0.014169            0.358           0.720
Wednesday                 -0.007690         0.014045           -0.548           0.584
Thursday                  -0.006148         0.013433           -0.458           0.647
L1.new_cases_pct          -0.003686         0.003186           -1.157           0.247
L1.stringency_diff         0.002941         0.001371            2.145           0.032
L1.hdax_pct                0.086830         0.137994            0.629           0.529
======================================================================================
```

```
Correlation matrix of residuals
                   new_cases_pct  stringency_diff  hdax_pct
new_cases_pct           1.000000         0.116427  0.047278
stringency_diff         0.116427         1.000000 -0.317209
hdax_pct                0.047278        -0.317209  1.000000
```

In [11]:

```
# Getting the summary of the results for the second sample period
results_2.summary()
```

Out[11]:

```
  Summary of Regression Results
==================================
Model:                         VAR
Method:                        OLS
Date:           Fri, 15, Oct, 2021
Time:                     09:12:42
--------------------------------------------------------------------
No. of Equations:         3.00000    BIC:                   -8.92778
Nobs:                     186.000    HQIC:                  -9.17534
Log likelihood:           101.225    FPE:                8.75020e-05
AIC:                     -9.34401    Det(Omega_mle):     7.71173e-05
--------------------------------------------------------------------
Results for equation new_cases_pct
=====================================================================================
                        coefficient       std. error           t-stat            prob
-------------------------------------------------------------------------------------
const                     -0.023543         0.120812           -0.195           0.845
Monday                    -0.382550         0.167677           -2.281           0.023
Tuesday                    0.739213         0.168927            4.376           0.000
Wednesday                  0.883410         0.175076            5.046           0.000
Thursday                   0.256418         0.171818            1.492           0.136
L1.new_cases_pct          -0.395347         0.068558           -5.767           0.000
L1.stringency_diff         0.029553         0.041845            0.706           0.480
L1.hdax_pct                8.249330         5.159317            1.599           0.110
=====================================================================================

Results for equation stringency_diff
=====================================================================================
                        coefficient       std. error           t-stat            prob
-------------------------------------------------------------------------------------
const                     -0.069367         0.212636           -0.326           0.744
Monday                    -0.228719         0.295121           -0.775           0.438
Tuesday                    0.448051         0.297322            1.507           0.132
Wednesday                  0.024684         0.308145            0.080           0.936
```

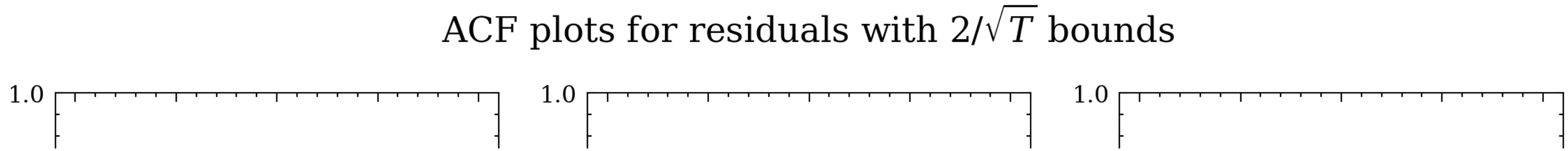
```
Thursday                   0.236416         0.302411            0.782           0.434
L1.new_cases_pct           0.207325         0.120667            1.718           0.086
L1.stringency_diff         0.169031         0.073650            2.295           0.022
L1.hdax_pct                5.021195         9.080712            0.553           0.580
=========================================================================================

Results for equation hdax_pct
=========================================================================================
                        coefficient       std. error           t-stat            prob
-----------------------------------------------------------------------------------------
const                      0.000257         0.001733            0.149           0.882
Monday                     0.001952         0.002405            0.812           0.417
Tuesday                    0.001832         0.002423            0.756           0.450
Wednesday                 -0.001355         0.002511           -0.539           0.590
Thursday                   0.000993         0.002464            0.403           0.687
L1.new_cases_pct           0.000728         0.000983            0.740           0.459
L1.stringency_diff         0.001220         0.000600            2.033           0.042
L1.hdax_pct               -0.056498         0.073995           -0.764           0.445
=========================================================================================

Correlation matrix of residuals
                   new_cases_pct  stringency_diff   hdax_pct
new_cases_pct           1.000000         0.192450 -0.008544
stringency_diff         0.192450         1.000000 -0.074398
hdax_pct               -0.008544        -0.074398  1.000000
```

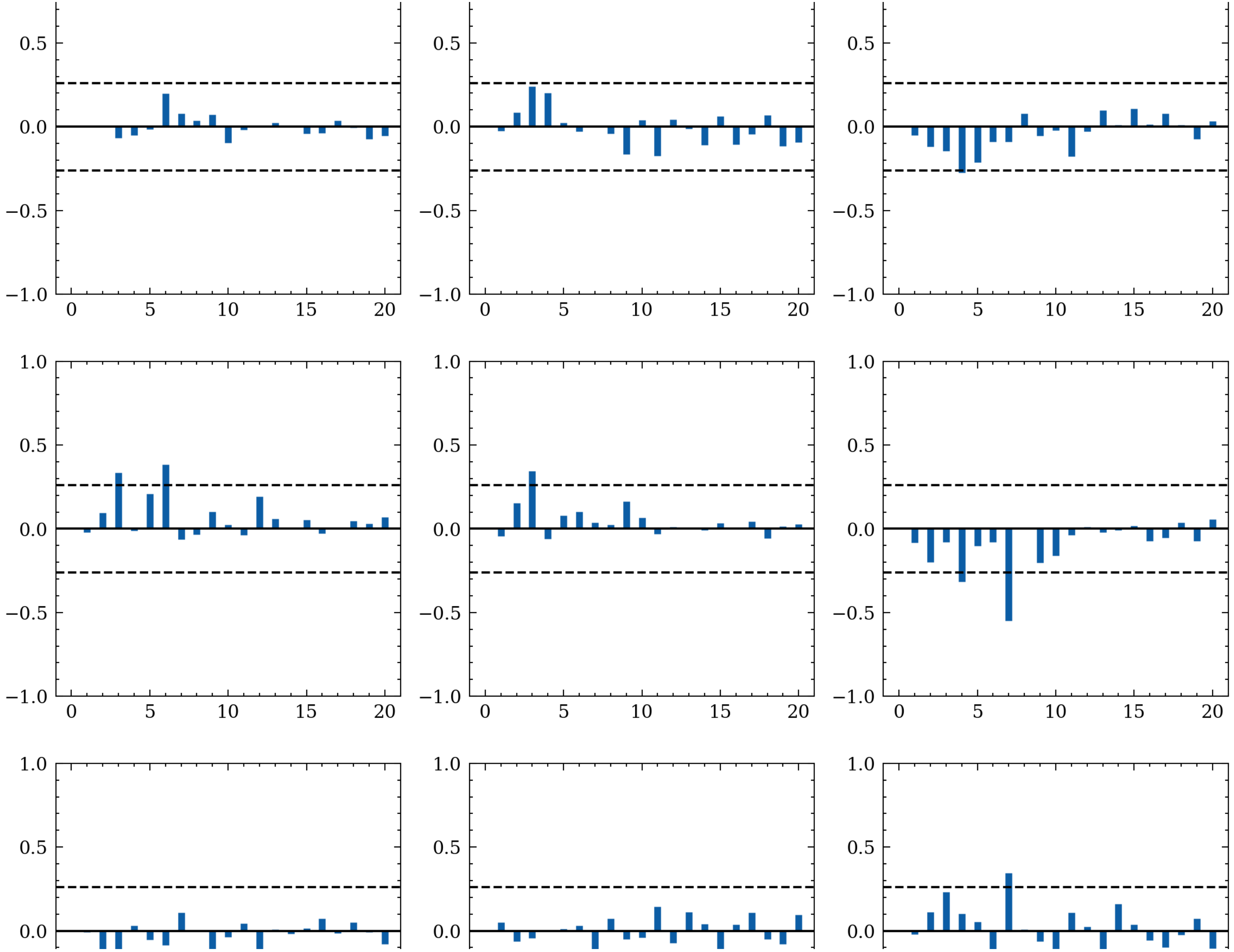
12. Plotting of Results

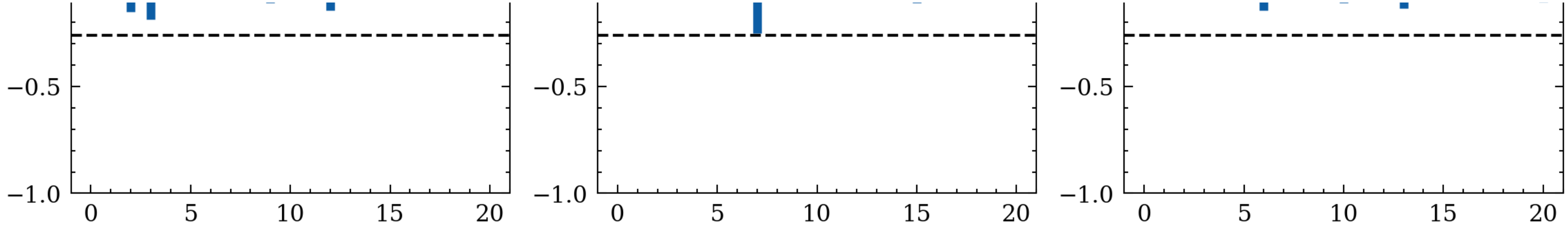
12.1 Plotting Results for first_1 sample timeframe

In [12]:

```python
plt.style.use(['science','no-latex'])
results_1_plot_acorr = results_1.plot_acorr(nlags=20, resid=True, linewidth=3)
results_1_plot_acorr.set_size_inches(8, 8)
results_1_plot_acorr.set_dpi(300)
results_1_plot_acorr.savefig("acf_plots_1.pdf")
```

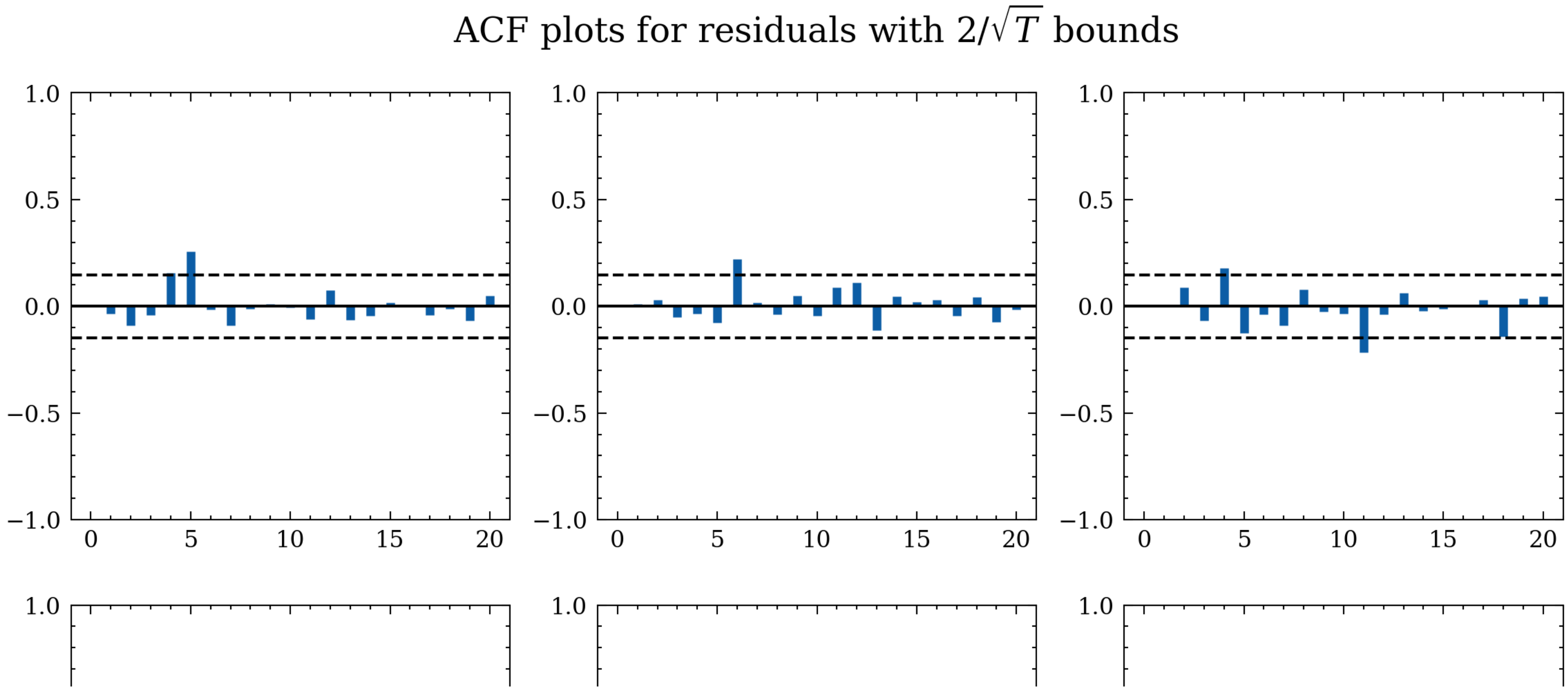


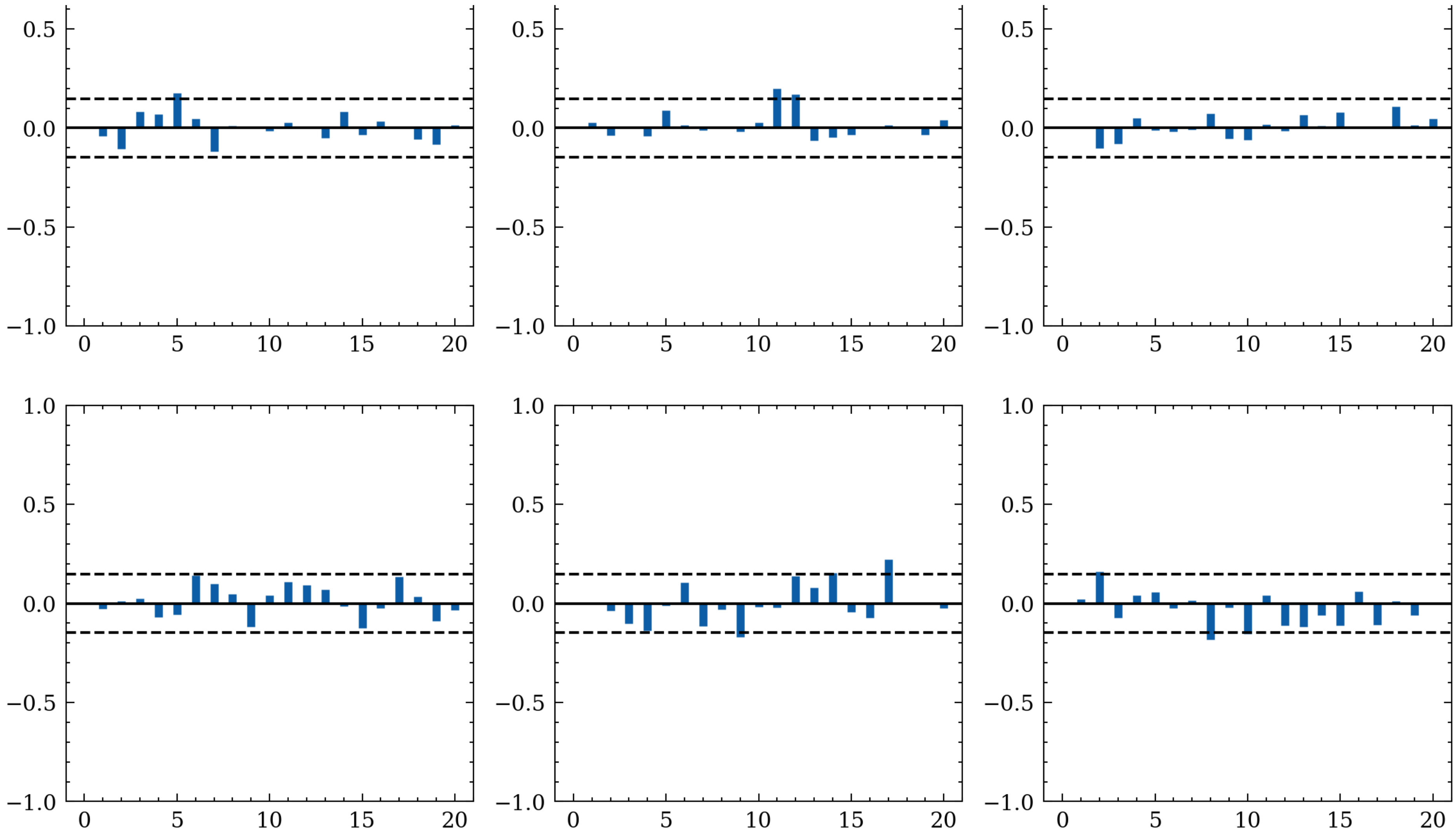

0.5
0.0
−0.5
−1.0
0
5
10
15
20
1.0



12.2 Plotting Results for second sample timeframe

In [13]:
```
plt.style.use(['science','no-latex'])
results_2_plot_acorr = results_2.plot_acorr(nlags=20, resid=True, linewidth=3)
results_2_plot_acorr.set_size_inches(8, 8)
results_2_plot_acorr.set_dpi(300)
results_2_plot_acorr.savefig("acf_plots_2.pdf")
```





Note: Shown are self (on the diagonal) and cross correlations!

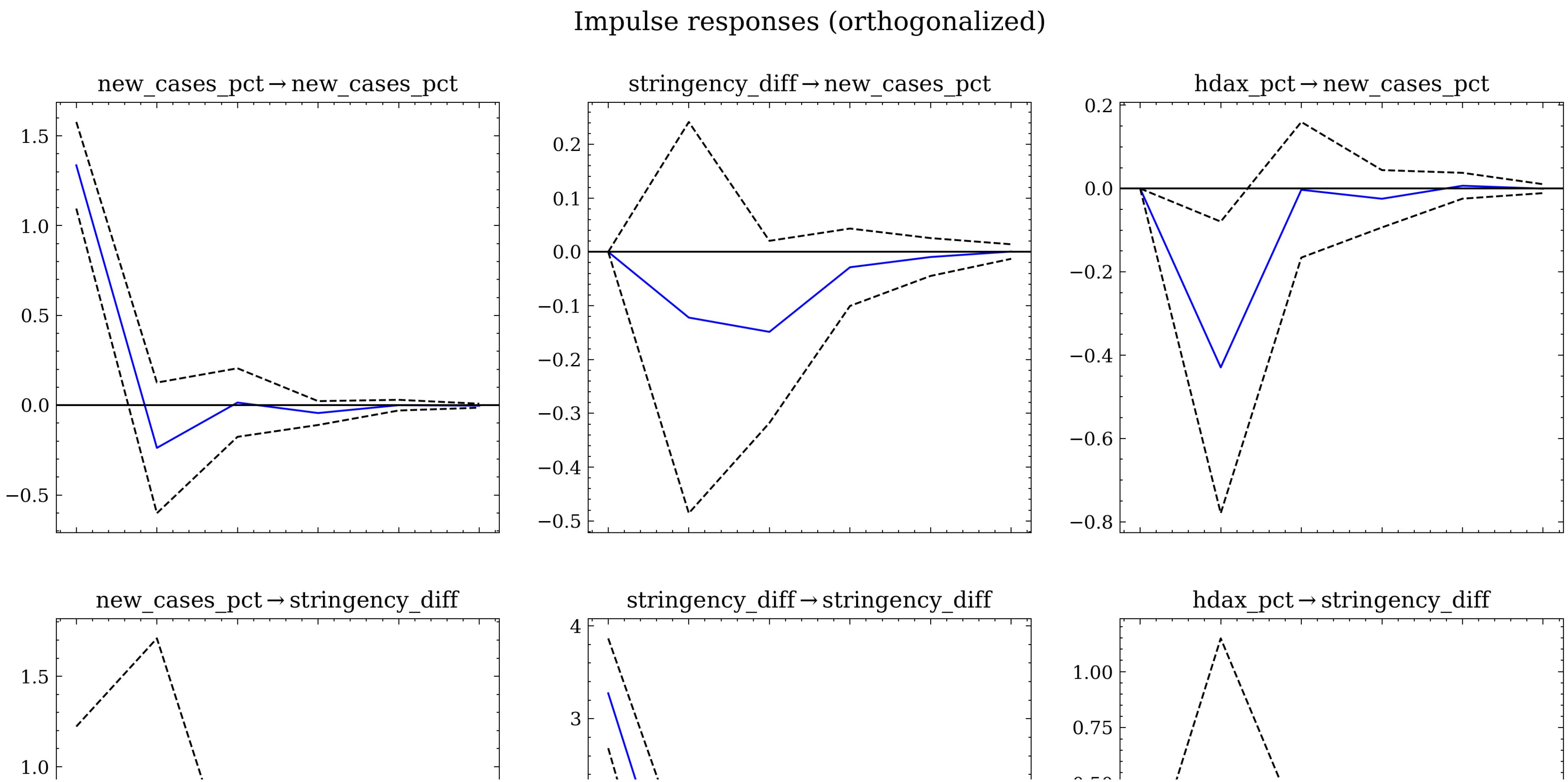
In order to remove the significant residual autocorrelations at low lags, it may help to fit a VAR(3) or VAR(4) model. Of course, this conflicts with choosing the model order on the basis of the model selection criteria. Thus, it has to be decided which criterion is given priority. It may be worth noting that a plot like that in Figure 4.2 may give a misleading picture of the overall significance of the residual autocorrelations because they are not asymptotically independent. Lütkepohl, Helmut, and Helmut Lütkepohl. New Introduction to Multiple Time Series Analysis, Springer Berlin / Heidelberg, 2007. ProQuest Ebook Central, http://ebookcentral.proquest.com/lib/unigiessen/detail.action?docID=6312046. Created from unigiessen on 2021-09-03 09:19:40.

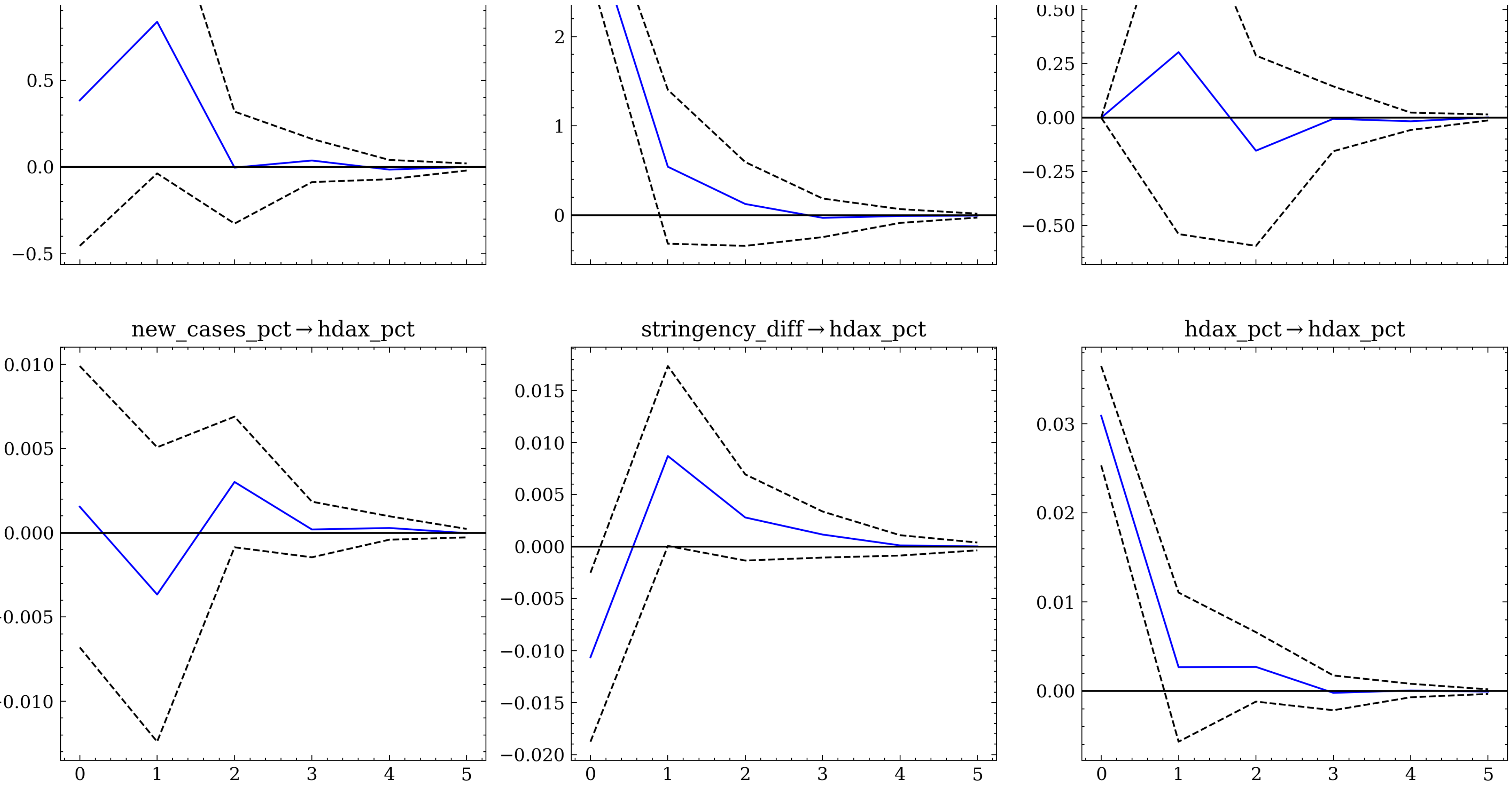
13. Impulse Response Functions (IRP)

13.1 IRP for first_1 sample timeframe

In [14]:
```
irf_1 = results_1.irf(5)
```

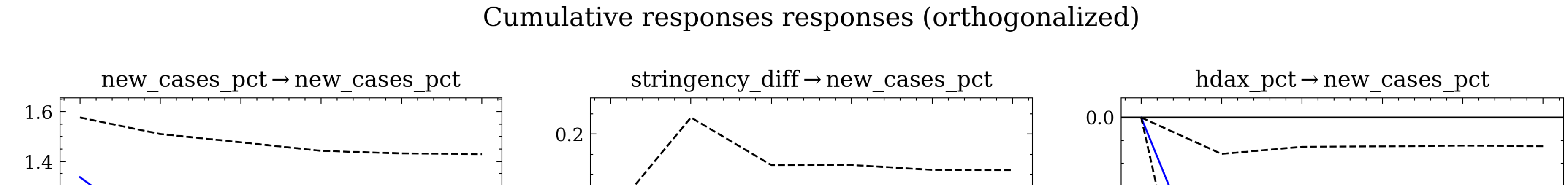
In [15]:
```
plt.style.use(['science','no-latex'])
irf_1_plot = irf_1.plot(orth=True)
irf_1_plot.set_size_inches(12.5, 12.5)
irf_1_plot.set_dpi(300)
irf_1_plot.savefig("irf_plots_1.pdf")
```

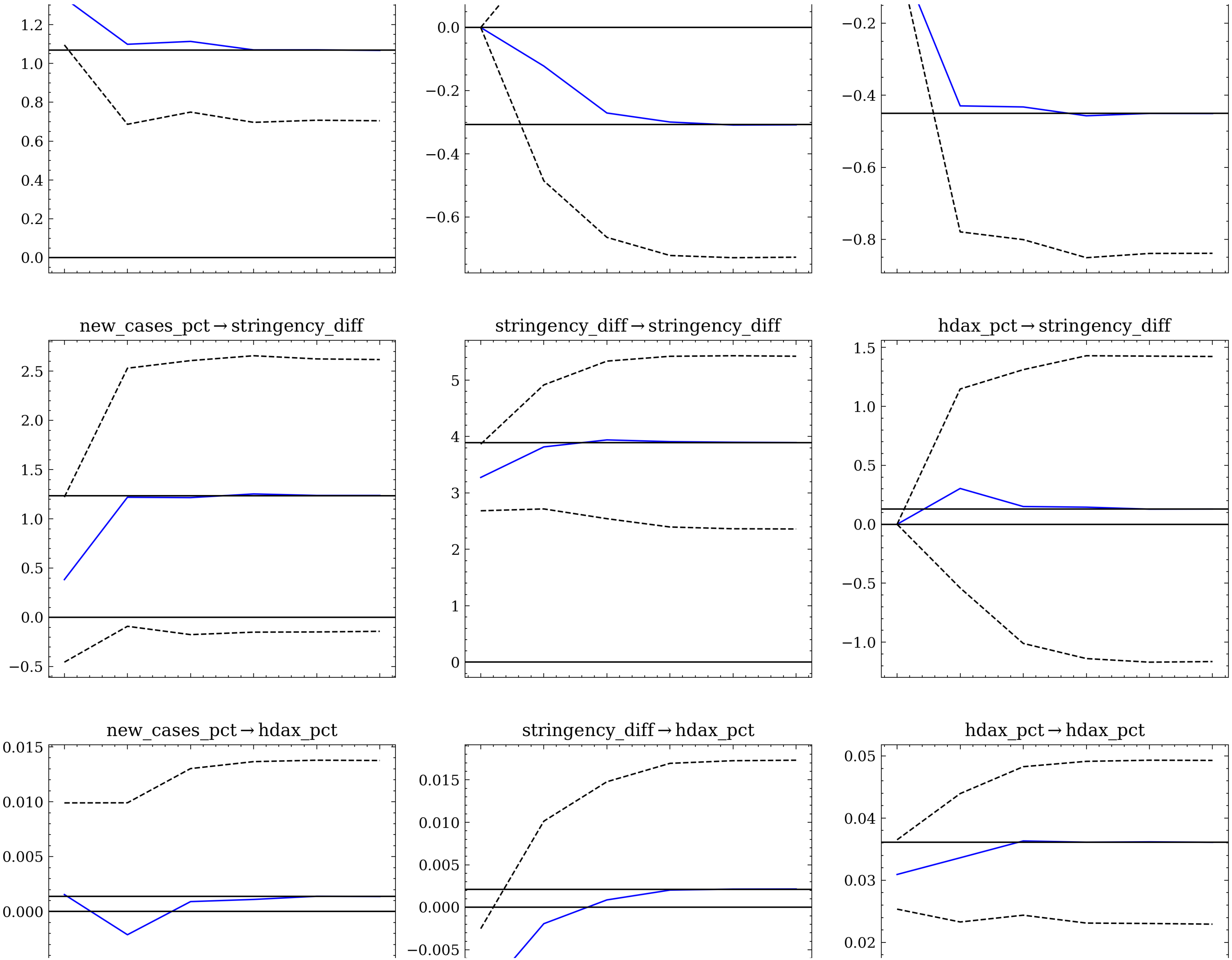


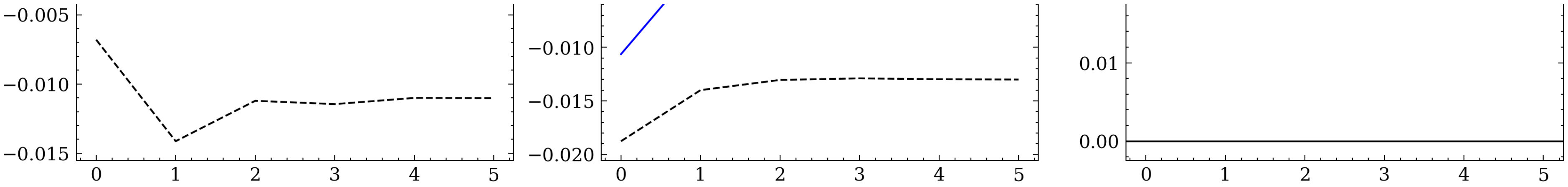


In [16]:

```
plt.style.use(['science','no-latex'])
irf_1_cum_plot = irf_1.plot_cum_effects(orth=True)
irf_1_cum_plot.set_size_inches(12.5, 12.5)
irf_1_cum_plot.set_dpi(300)
irf_1_cum_plot.savefig("irf_cum_plots_1.pdf")
```



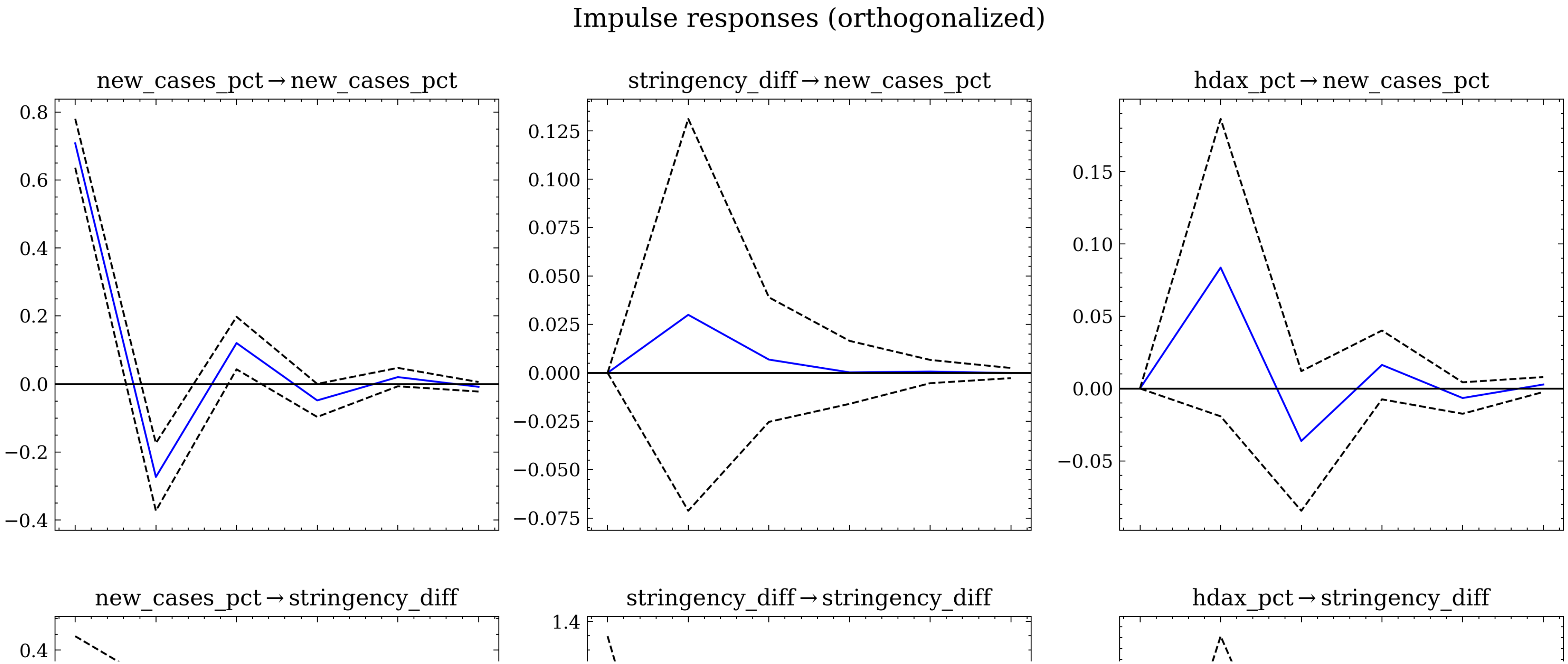

1.2
1.0
0.8
0.6
0.4
0.2
0.0
0.0
−0.2
−0.4
−0.6
−0.2
−0.4
−0.6
−0.8
new_cases_pct → stringency_diff
stringency_diff → stringency_diff
hdax_pct → stringency_diff
2.5
2.0
1.5
1.0
0.5
0.0
−0.5
5
4
3
2
1
0
1.5
1.0
0.5
0.0
−0.5
−1.0
new_cases_pct → hdax_pct
stringency_diff → hdax_pct
hdax_pct → hdax_pct
0.015
0.010
0.005
0.000
0.015
0.010
0.005
0.000
−0.005
0.05
0.04
0.03
0.02

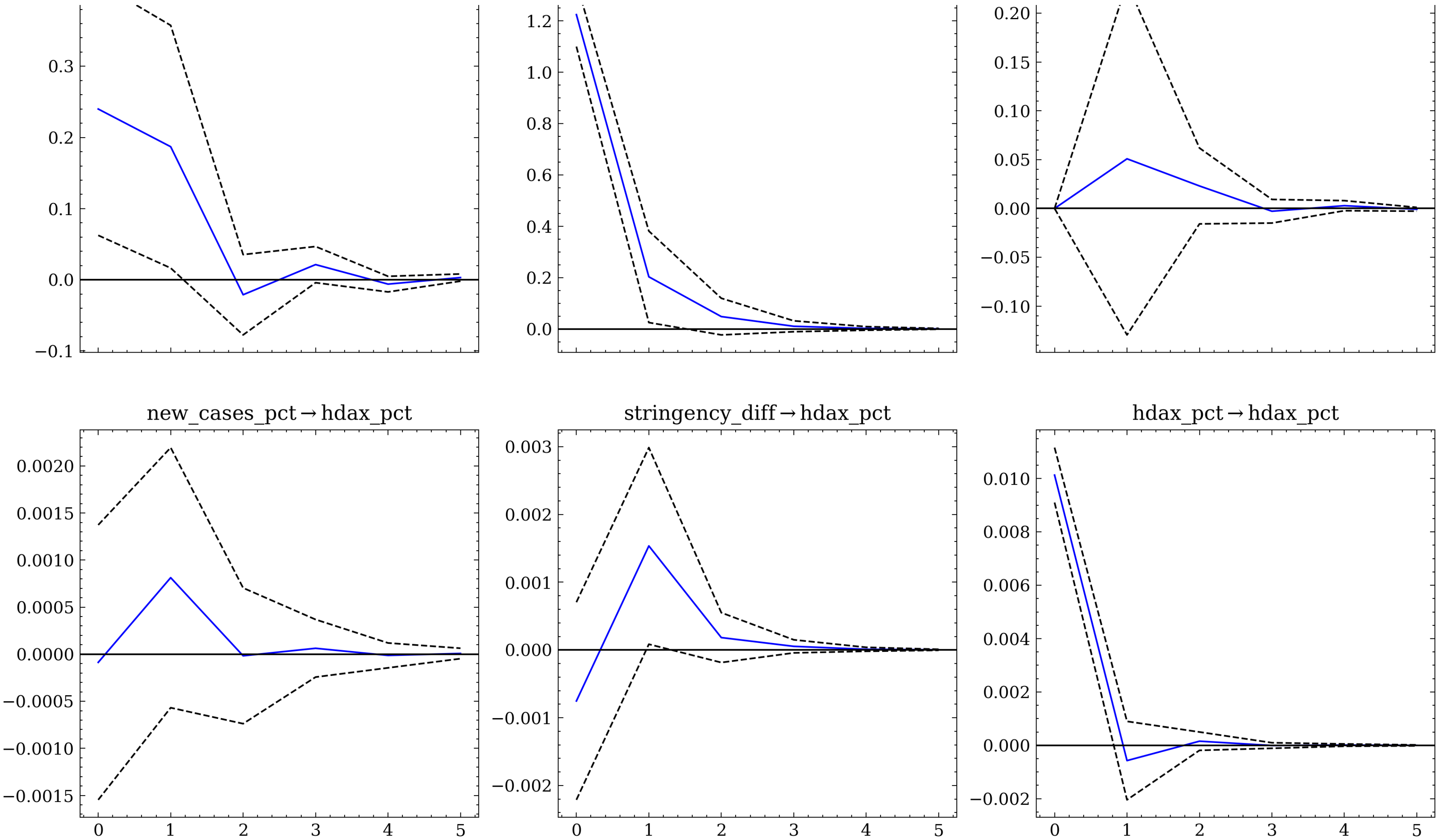


13.2 IRP for second sample timeframe

In [17]:
```
irf_2 = results_2.irf(5)
```

In [18]:
```
plt.style.use(['science','no-latex'])
irf_2_plot = irf_2.plot(orth=True)
irf_2_plot.set_size_inches(12.5, 12.5)
irf_2_plot.set_dpi(300)
irf_2_plot.savefig("irf_plots_2.pdf")
```

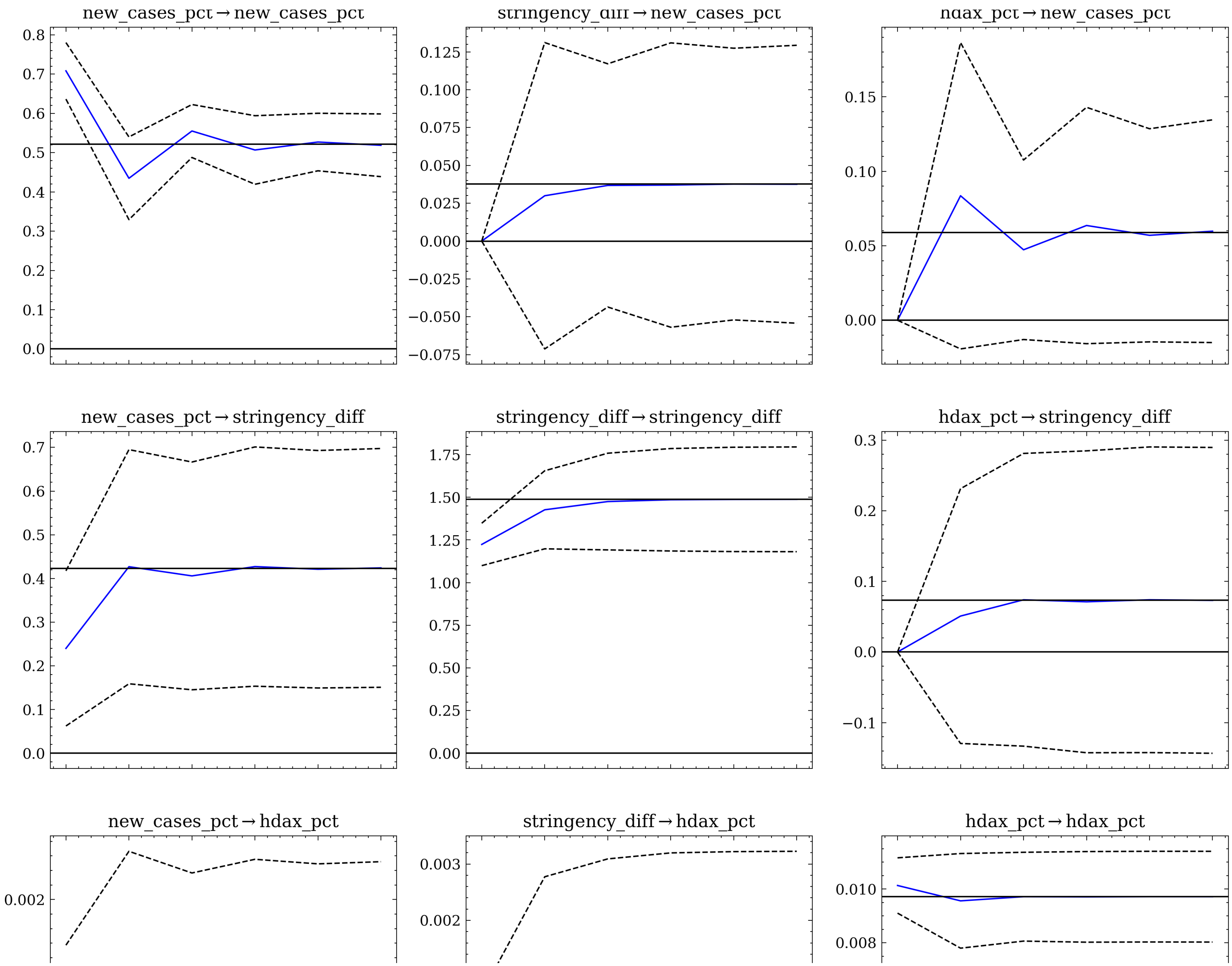


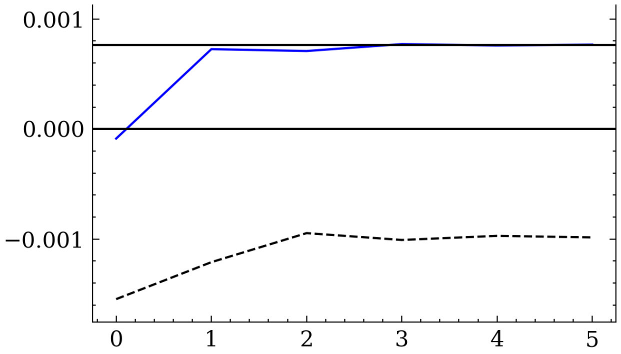


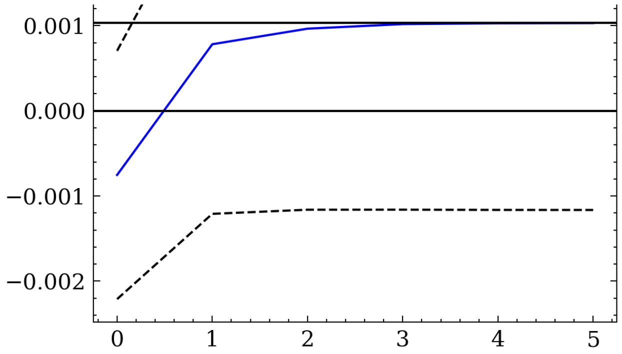
In [19]:

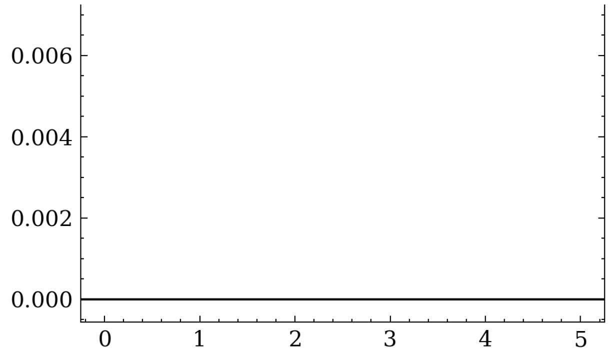
```
plt.style.use(['science','no-latex'])
irf_2_cum_plot = irf_2.plot_cum_effects(orth=True)
irf_2_cum_plot.set_size_inches(12.5, 12.5)
irf_2_cum_plot.set_dpi(300)
irf_2_cum_plot.savefig("irf_cum_plots_2.pdf")
```

Cumulative responses responses (orthogonalized)


new_cases_pct → new_cases_pct
stringency_diff → new_cases_pct
hdax_pct → new_cases_pct
new_cases_pct → stringency_diff
stringency_diff → stringency_diff
hdax_pct → stringency_diff
new_cases_pct → hdax_pct
stringency_diff → hdax_pct
hdax_pct → hdax_pct







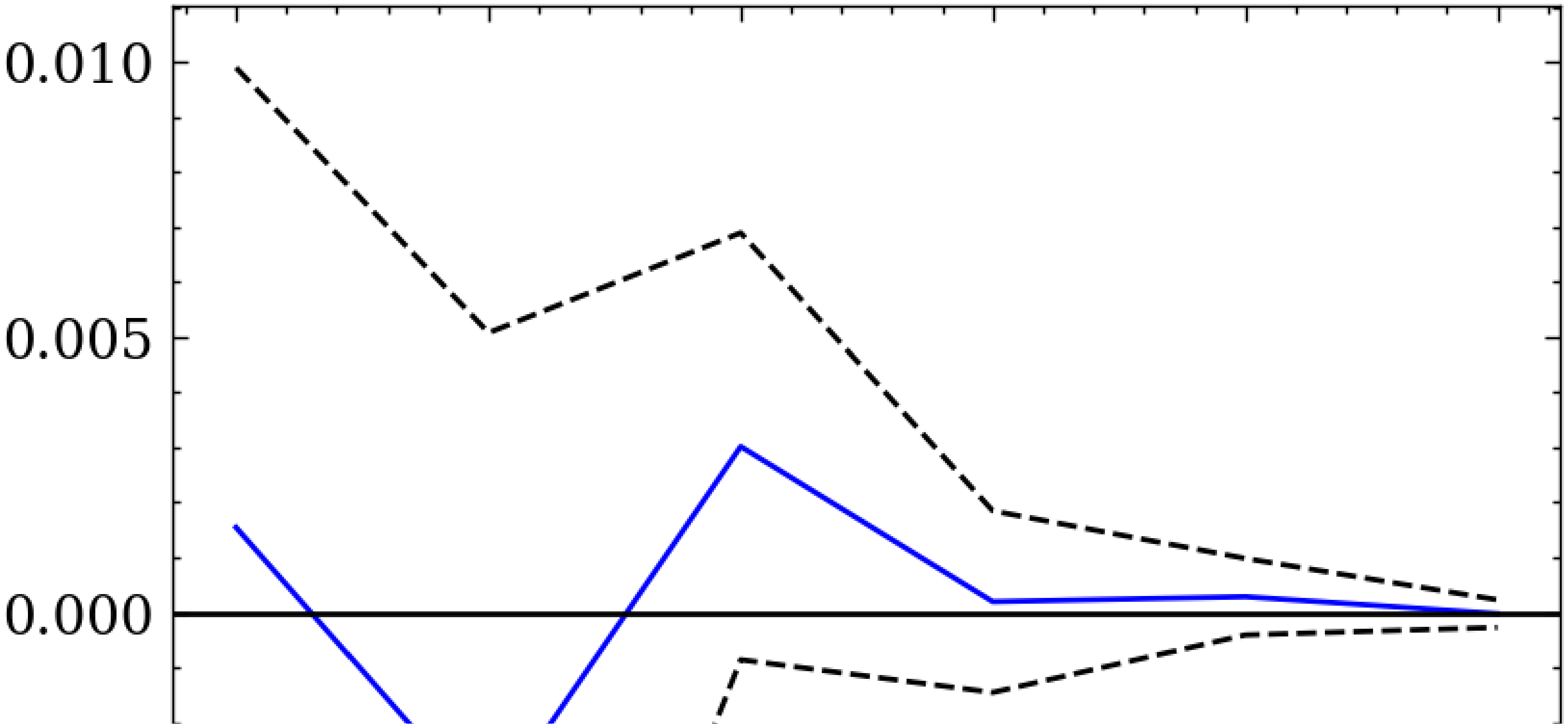
In [20]:

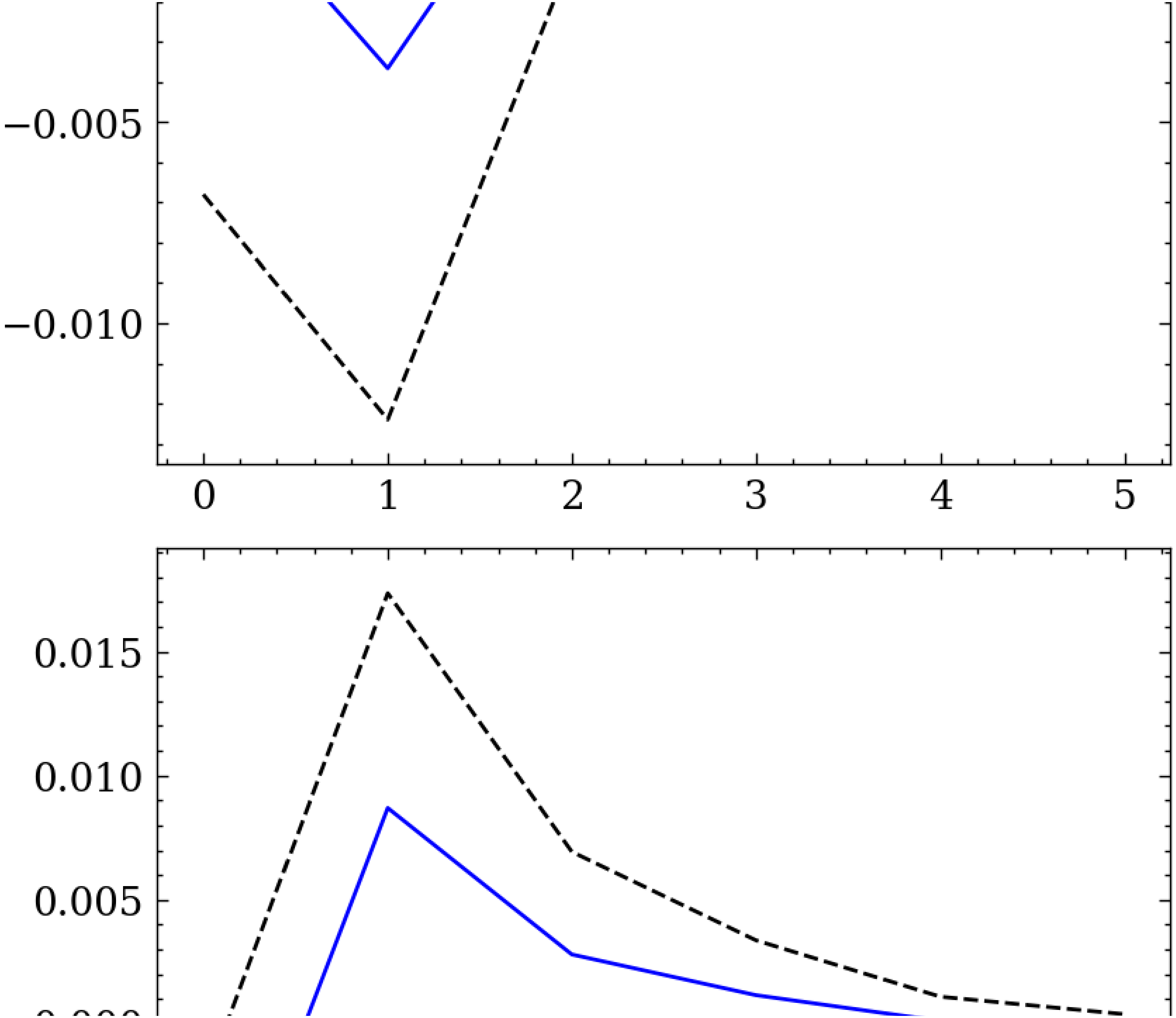
```
# single cumulative plots

#irf_1_1_plot_cumulative_cases = irf_1_1.plot_cum_effects(impulse='new_cases_pct',
 #                                                        response="hdax_pct",
  #                                                       orth=True)
#
#irf_2_plot_cumulative_cases = irf_2.plot_cum_effects(impulse='new_cases_pct',
 #                                                        response="hdax_pct",
  #                                                       orth=True)
#
#irf_2_plot_cumulative_stringency = irf_2.plot_cum_effects(impulse='stringency_diff',
 #                                                        response="hdax_pct",
  #                                                       orth=True)
```

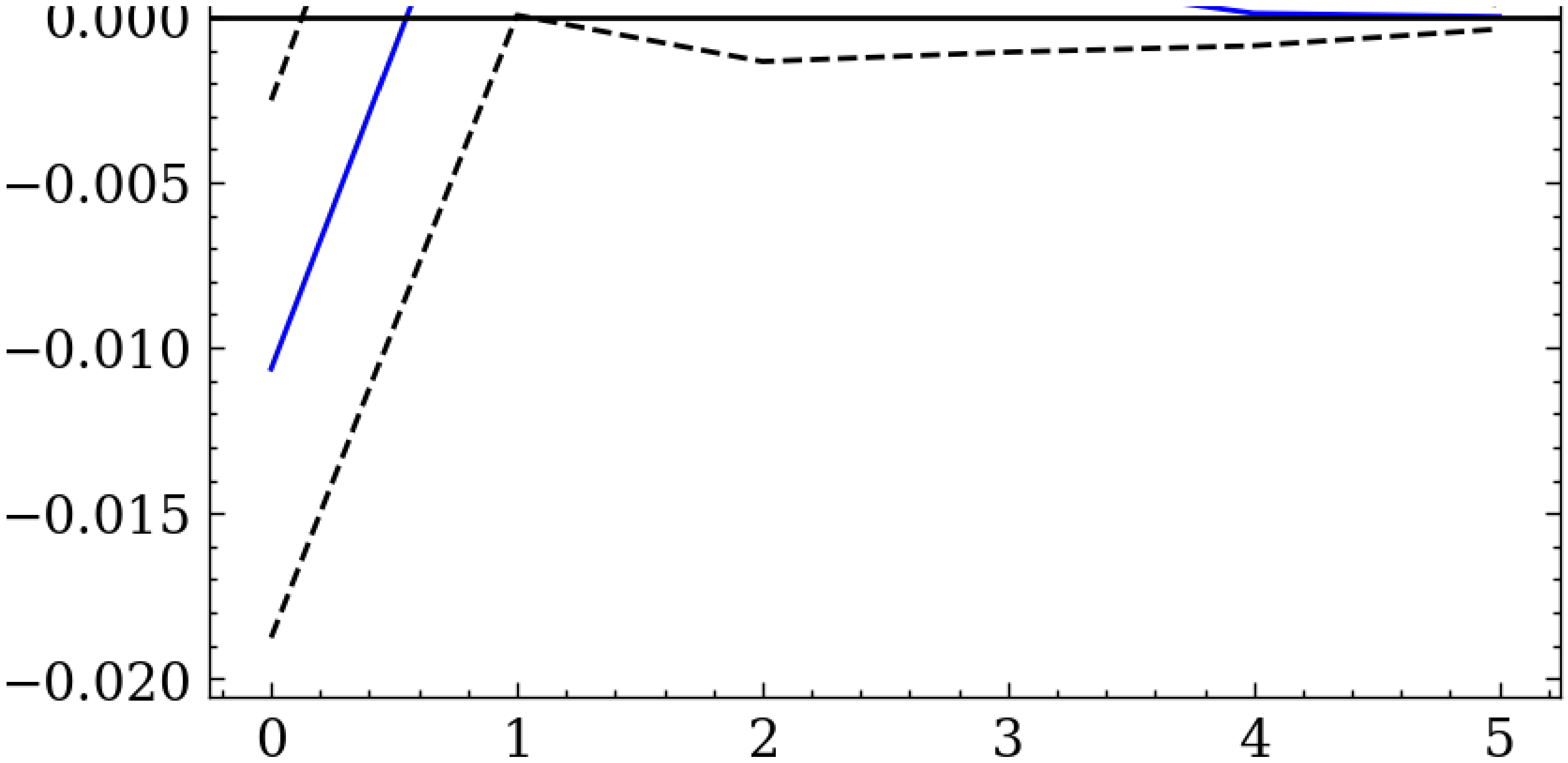
In [21]:

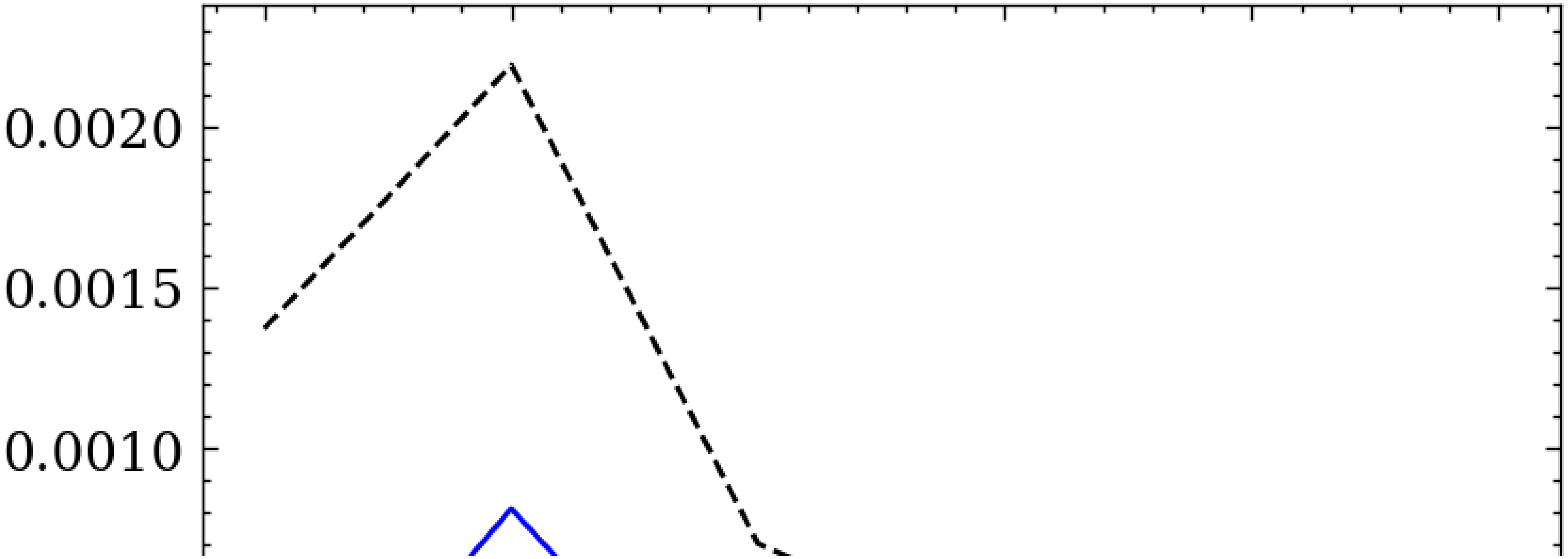
```
# exporting single irf plots
irf_1_plot_new_cases = irf_1.plot(impulse='new_cases_pct' ,
                                   response="hdax_pct",
                                   orth=True)
irf_1_plot_new_cases.set_size_inches(4, 4)
irf_1_plot_new_cases.set_dpi(200)
irf_1_plot_new_cases.suptitle("")
irf_1_plot_new_cases.get_axes()[0].set_title("")
irf_1_plot_new_cases.savefig("irf_1_plot_new_cases.pdf")

irf_1_plot_stringency = irf_1.plot(impulse='stringency_diff',
                                    response="hdax_pct",
                                    orth=True, )
irf_1_plot_stringency.set_size_inches(4, 4)
irf_1_plot_stringency.set_dpi(200)
irf_1_plot_stringency.suptitle("")
```

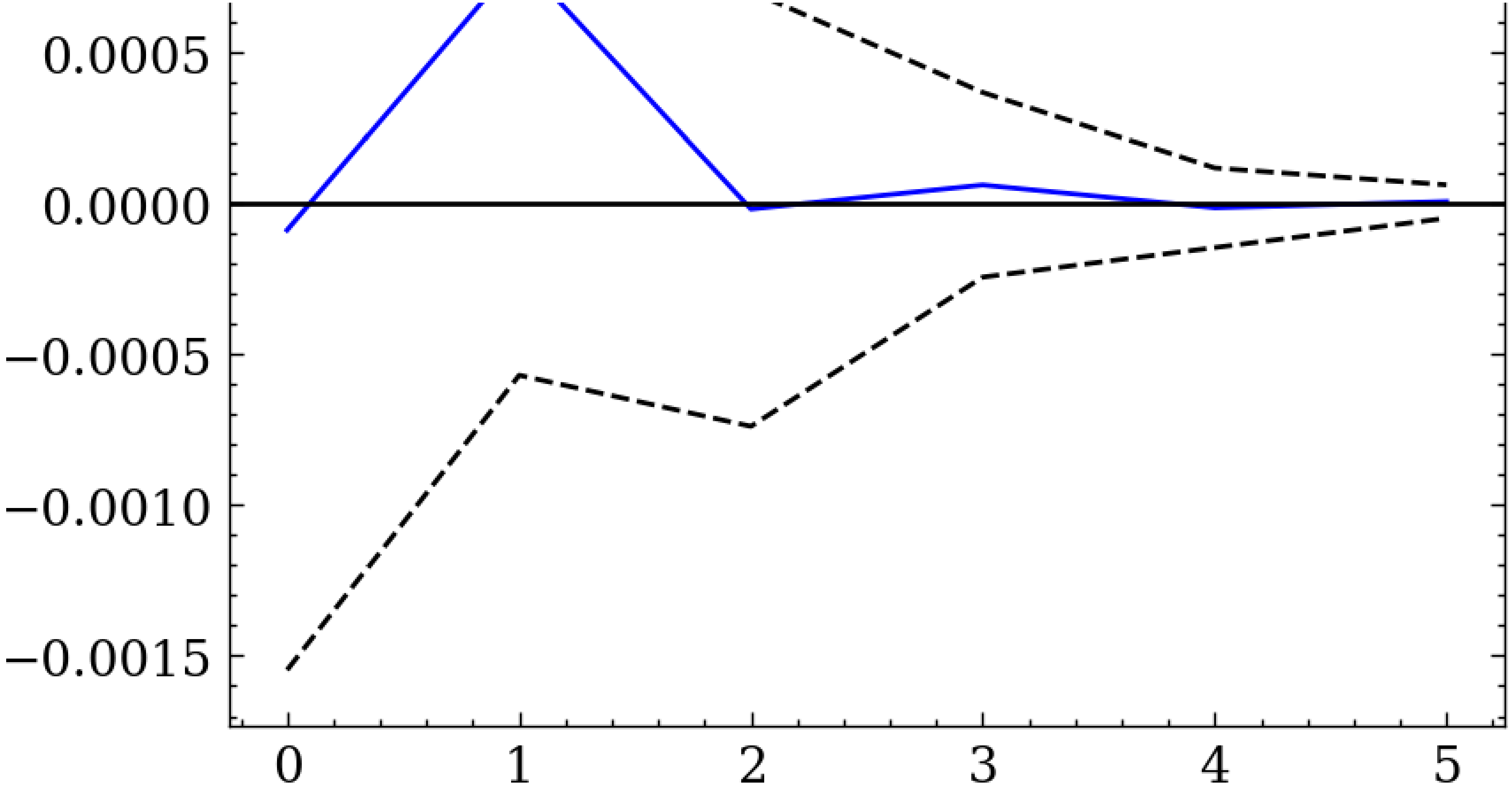
```
irf_1_plot_stringency.get_axes()[0].set_title("")
irf_1_plot_stringency.savefig("irf_1_plot_stringency.pdf")

irf_2_plot_cases = irf_2.plot(impulse='new_cases_pct',
                              response="hdax_pct",
                              orth=True)
irf_2_plot_cases.set_size_inches(4, 4)
irf_2_plot_cases.set_dpi(200)
irf_2_plot_cases.suptitle("")
irf_2_plot_cases.get_axes()[0].set_title("")
irf_2_plot_cases.savefig("irf_2_plot_new_cases.pdf")

irf_2_plot_stringency = irf_2.plot(impulse='stringency_diff',
                                   response="hdax_pct",
                                   orth=True)
irf_2_plot_stringency.set_size_inches(4, 4)
irf_2_plot_stringency.set_dpi(200)
irf_2_plot_stringency.suptitle("")
irf_2_plot_stringency.get_axes()[0].set_title("")
irf_2_plot_stringency.savefig("irf_2_plot_stringency.pdf")
```

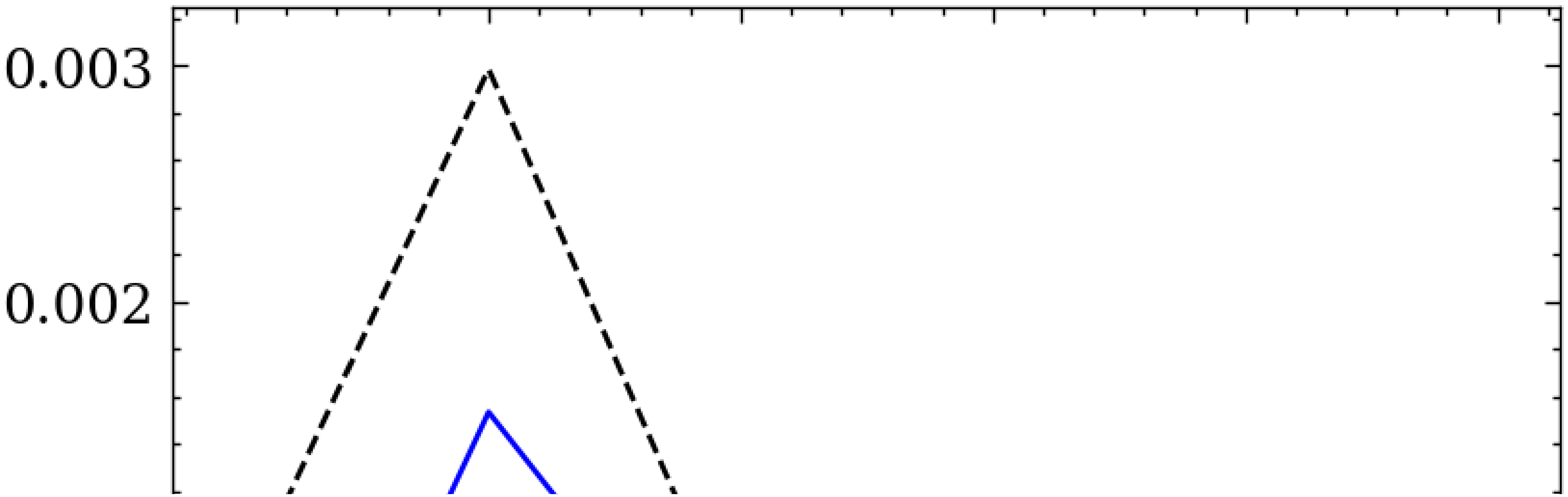


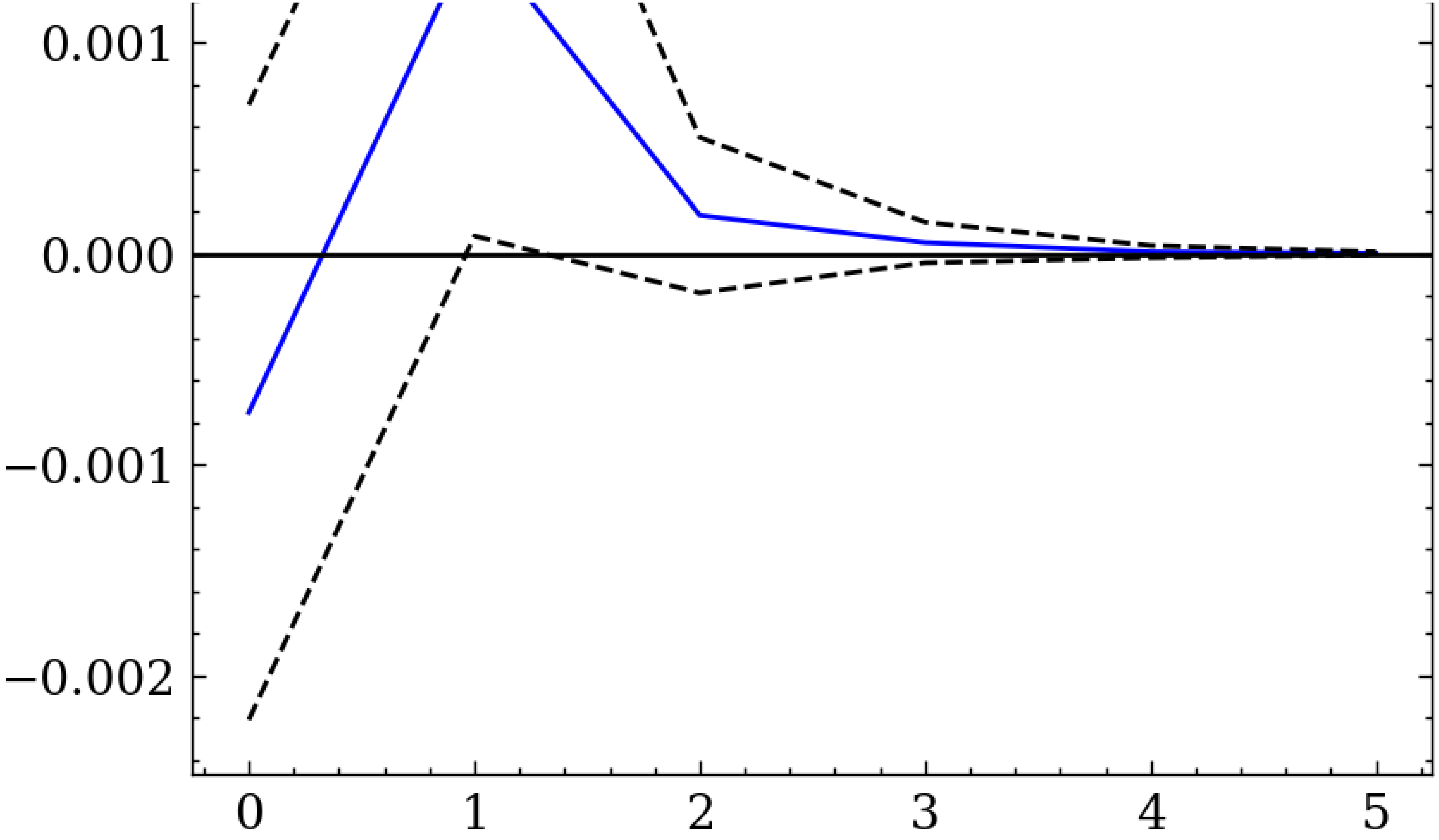

−0.005
−0.010
0
1
2
3
4
5
0.015
0.010
0.005


0.000
−0.005
−0.010
−0.015
−0.020
0
1
2
3
4
5


0.0020
0.0015
0.0010


0.0005
0.0000
−0.0005
−0.0010
−0.0015
0
1
2
3
4
5


0.003
0.002



14. Statistical Tests

14.1 Granger Causality Tests

- H0: “the causing variables do not Granger-cause the remaining variables of the system”
- H1: “the causing variables is Granger-causal for the remaining variables”

Granger causality test does not explain some aspects of the VAR:

- It does not give the sign of the effect, we do not know if it is positive or negative

- It does not show how long the effectlasts for.
- It does not provide evidence of whether this effect is direct or indirect.

14.1.1 Granger Causality tests for first_1 sample

In [22]:

```
#Testing block exogenity of new cases and stringency on stock returns
causality_1_block = results_1.test_causality('hdax_pct', ['new_cases_pct', 'stringency_diff'], kind='f', signif=0.05)

#Testing granger causality of cases on stock returns
causality_1_cases = results_1.test_causality('hdax_pct', 'new_cases_pct', kind='wald', signif=0.05)

#Testing granger causality of stringency on stock returns
causality_1_stringency = results_1.test_causality('hdax_pct', 'stringency_diff', kind='wald', signif=0.05)
```

In [23]:

```
causality_1_block.summary()
```

Out[23]:

Granger causality F-test. H_0: ['new_cases_pct', 'stringency_diff'] do not Granger-cause hdax_pct. Conclusion: fail to reject H_0 at 5% significance level.

Test statistic	Critical value	p-value	df
2.916	3.055	0.057	(2, 153)

In [24]:

```
causality_1_cases.summary()
```

Out[24]:

Granger causality Wald-test. H_0: new_cases_pct does not Granger-cause hdax_pct. Conclusion: fail to reject H_0 at 5% significance level.

Test statistic	Critical value	p-value	df
1.339	3.841	0.247	1

In [25]:

```
causality_1_stringency.summary()
```

Out[25]:

Granger causality Wald-test. H_0: stringency_diff does not Granger-cause hdax_pct. Conclusion: reject H_0 at 5% significance level.

Test statistic	Critical value	p-value	df
4.600	3.841	0.032	1

14.1.2 Granger Causality tests for second sample

In [26]:

```
#Testing block exogenity of new cases and stringency on stock returns
causality_2_block = results_2.test_causality('hdax_pct', ['new_cases_pct', 'stringency_diff'], kind='f', signif=0.05)

#Testing granger causality of cases on stock returns
causality_2_cases = results_2.test_causality('hdax_pct', 'new_cases_pct', kind='wald', signif=0.05)

#Testing granger causality of stringency on stock returns
causality_2_stringency = results_2.test_causality('hdax_pct', 'stringency_diff', kind='wald', signif=0.05)
```

In [27]:

```
causality_2_block.summary()
```

Out[27]:

Granger causality F-test. H_0: ['new_cases_pct', 'stringency_diff'] do not Granger-cause hdax_pct. Conclusion: fail to reject H_0 at 5% significance level.

Test statistic	Critical value	p-value	df
2.567	3.013	0.078	(2, 534)

In [28]:

```
causality_2_cases.summary()
```

Out[28]:

Granger causality Wald-test. H_0: new_cases_pct does not Granger-cause hdax_pct. Conclusion: fail to reject H_0 at 5% significance level.

Test statistic	Critical value	p-value	df

0.5476	3.841	0.459	1

In [29]:

```
causality_2_stringency.summary()
```

Out[29]:

Granger causality Wald-test. H_0: stringency_diff does not Granger-cause hdax_pct. Conclusion: reject H_0 at 5% significance level.

Test statistic	Critical value	p-value	df
4.133	3.841	0.042	1

14.2 Normality Tests

- H0: "The data is generated by a Gaussian-distributed process"
- H1: "The data is not generated by a Gaussian-distributed process"

In [30]:

```
#Testing normality of first_1 sample
normality_1 = results_1.test_normality()

#Testing normality of second sample
normality_2 = results_2.test_normality()
```

In [31]:

```
normality_1.summary()
```

Out[31]:

normality (skew and kurtosis) test. H_0: data generated by normally-distributed process. Conclusion: reject H_0 at 5% significance level.

Test statistic	Critical value	p-value	df
846.6	12.59	0.000	6

In [32]:

```
normality_2.summary()
```

normality (skew and kurtosis) test. H_0: data

Out[32]:

generated by normally-distributed process. Conclusion: reject H_0 at 5% significance level.

Test statistic	Critical value	p-value	df
5574.	12.59	0.000	6

14.3 Residual Autocorrelation Tests

Note:

We use the Portmanteau test with reasonably large number of lags to test for the overall significance of the residual autocorrelations up to lag h. "For practical purposes, it is important to remember that the χ 2 - approximation to the distribution of the test statistic may be misleading for small values of h ." and "For large h, the degrees of freedom in the auxiliary regression model will be exhausted". See Lütkepohl: In contrast to the portmanteau tests which should be used for reasonably large h only, the LM tests are more suitable for small values of h . For large h , the degrees of freedom in the auxiliary regression model will be exhausted Lütkepohl, Helmut, and Helmut Lütkepohl. New Introduction to Multiple Time Series Analysis, Springer Berlin / Heidelberg, 2007. ProQuest Ebook Central, http://ebookcentral.proquest.com/lib/unigiessen/detail.action?docID=6312046. Created from unigiessen on 2021-09-03 08:45:14.

However: (Verteilungsannahme gilt nicht, falls verzögerte endogene und exogene Variable in der Regressionsgleichung auftauchen (Dezhbakhsh 1990): "The experiments also indicate that the portmanteau test is inadequate when applied to dynamic linear models with exogenous regressors.")

There is no possibility yet to test for residual autocorrelation in multivariate systems using the Bruesch Godfrey Lagrange Multiplier residual serial correlation tests in the python package statsmodels. Instead, I used the portmanteau test for residual autocorrelation. This is, however, already a limitation on the validity of the results according to Maddala (2001) "Introduction to Econometrics (3d edition), ch 6.7, and 13. 5 p 528. Maddala who laments the widespread use of this test, and instead considers as appropriate the "Langrange Multiplier" test of Breusch and Godfrey.

- H0: "The timeseries are white noise (absence of significant residual autocorrelations)"
- H1: "The timeseries are not white noise (presence of significant residual autocorrelations)"

In [33]:

```
#Testing for residual autocorrelation in the first_1 sample
whiteness_1 = results_1.test_whiteness(nlags=int(results_1.nobs*0.25), signif=0.05, adjusted=False)
```

```python
#Testing for residual autocorrelation in the second sample
whiteness_2 = results_2.test_whiteness(nlags=int(results_2.nobs*0.25), signif=0.05, adjusted=False)
```

In [34]:

```python
whiteness_1.summary()
```

Out[34]:

Portmanteau-test for residual autocorrelation. H_0: residual autocorrelation up to lag 14 is zero. Conclusion: fail to reject H_0 at 5% significance level.

Test statistic	Critical value	p-value	df
115.2	143.2	0.530	117

In [35]:

```python
whiteness_2.summary()
```

Out[35]:

Portmanteau-test for residual autocorrelation. H_0: residual autocorrelation up to lag 46 is zero. Conclusion: fail to reject H_0 at 5% significance level.

Test statistic	Critical value	p-value	df
416.4	452.9	0.337	405

In [ ]: