Some Standardized shocks to illustrate the main macroeconomic features of the model

This notebook performs a set of standard macroeconomic shocks on a model and displays results. It can be used to verify the good functioning of the model and also illustrates nicely how to perform simulations on World Bank models in python using the `modelflow` package.

The simulations performed are:

- 5 separate stimulus shocks equal to 1 percent of *ex ante* GDP. These can be used to compare fiscal multipliers and the impacts of different kinds of stimulus on GDP, potential GDP, consumption and inflation
  - a 1 percent of GDP decrease in indirect taxes
  - a 1 percent of GDP decrease in direct taxes
  - a 1 percent of GDP increase in government spending on goods and services
  - a 1 percent of GDP increase in government spending on investment goods
  - a 1 percent of GDP increase in government spending on transfers to households
- 4 non-fiscal shocks were also run
  - a temporary one-year 1 percent increase in the monetary policy interest rate;
  - a permanent 10 percent depreciation of the currency
  - a permanent one percent increase in total factor productivity;
  - a permanent $20 increase in the price of crude oil

Note: This Notebook is designed to be run on any World Bank model. To customize the notebook to run on a different model, the string variable (`Cty`), which is defined in section 1.1 will have to be changed to take the value of the Mnemonic of the country to be simulated. Similarly the location of the file containing the model may have to be revised. While the World Bank mnemonics are the same across countries, not all countries report all variables. As a result for some models, some variable names (notably those of shocked variables or the expenditure variables being held constant) may need to be revised. Otherwise the Notebook should run without change on any World Bank model.

Set up python environment and load model object

To work with modelflow we must first import the python libraries that we wish to work with and then instantiate the model object, which we have chose to call `themodel` ;

In [1]:
```
from modelclass import model
```

In [2]:
```
#Jupyter notebook code that improves the look of the executed notebook
%load_ext autoreload
%autoreload 2
```

In [3]:
```
#Set this variable to the three-letter ISO of the country whose model is being simu
Cty="NPL"
# Models downloaded from the World Bank web site using the model.download_github_re
# executing this file from the local version of the file stored on their computer s
filepath=f'data/{Cty}'
themodel,bline = model.modelload(filepath,run=True,keep='Baseline')
```

```
Zipped file read:  C:\Users\wb268970\OneDrive - WBG\Ldrive\MFM\wbg-Modelflow\MFMod_M
odelFlow\Nepal\data\NPL.pcim
Will start solving: NPL
New data or transpile_reset
Create compiled solving function for NPL
ljit=False stringjit=False  transpile_reset=False  hasattr(self, f"pro_{jitname}")=F
alse
now makelos makes a sim solvefunction
2023 Solved in 6 iterations
2024 Solved in 6 iterations
2025 Solved in 6 iterations
2026 Solved in 6 iterations
2027 Solved in 6 iterations
2028 Solved in 6 iterations
2029 Solved in 6 iterations
2030 Solved in 6 iterations
NPL solved
```

Prepare the simulations

For each shock a separate dataframe is created. Each of these dataframes is given a name that evokes the shock to be performed. Then each dataframe is modified to reflect the shock that is to be performed.

Following the creation of the dataframes the shocks will be performed and the results stored using the `keep=` syntax of model flow.

Fiscal policy shocks

If necessary, the two lines below can be uncommented in order to generate a list of all variables in the model that start GGEXP (general government expenditure) and GGREV (general governent revenues) and that end CN (millions of current local currency units).

In [4]:
```
themodel['???GGEXP*CN'].des #Uncomment to get list of mnemonics and descriptions fo
themodel['???GGREV*CN'].des #Uncomment to get list of mnemonics and descriptions fo
```

```
NPLGGEXPCAPTCN : Capital_Expenditures
NPLGGEXPCRNTCN : Current_expenditures
NPLGGEXPGCAPCN : Social_Service_Capital_Grants_Expenditure
NPLGGEXPGCURCN : Social_Service_Recurrent_Grants_Expenditure
NPLGGEXPGGCACN : Grant_to_Other_Levels_of_Government_Capital
NPLGGEXPGGFTCN : Grant_to_Other_Levels_of_Government_Fiscal_Transfers
NPLGGEXPGGOTCN : Grant_to_Other_Levels_of_Government_Others
NPLGGEXPGGOVCN : Grant_to_Other_Levels_of_Government
NPLGGEXPGITLCN : Grants_Expenditure_Intl_Org_Membership
NPLGGEXPGNFSCN : Goods_and_services
NPLGGEXPGRNTCN : Grants_Expenditure
NPLGGEXPGSSVCN : Social_Service_Grants_Expenditure
NPLGGEXPINTDCN : Domestic_debt_interest_payments
NPLGGEXPINTECN : External_debt_interest_payments
NPLGGEXPINTPCN : Interest_payments
NPLGGEXPOTHRCN : All_other_Expenditures_not_classified_elsewhere
NPLGGEXPSUBSCN : Subsidies
NPLGGEXPTOTHCN : Other_Current_Expenditure
NPLGGEXPTOTLCN : Total_expenditure
NPLGGEXPTOTSCN : Social_Securities
NPLGGEXPTPNSCN : Pension_Employment_Related_to_Social_Benefit_of_Employees
NPLGGEXPTRNSCN : Current_transfers
NPLGGEXPTSOCCN : Social_Assistance
NPLGGEXPTSSCCN : Total_Social_Securities_Expenditure
NPLGGEXPWAGECN : Compensation_of_employees
NPLGGREVDCITCN : Taxes_on_income_of_enterprises_and_corporations
NPLGGREVDINVCN : Taxes_on_investment_and_other_income
NPLGGREVDPITCN : Taxes_on_sole_traders_and_individual_income
NPLGGREVDRCTCN : Direct_taxes
NPLGGREVGEXCCN : Excise
NPLGGREVGNFSCN : Taxes_on_goods_and_services
NPLGGREVGOTHCN : Taxes_on_goods_and_services_others
NPLGGREVGRNTCN : Grants_revenues
NPLGGREVGVATCN : VAT
NPLGGREVNONTCN : Non_tax_revenues
NPLGGREVOTHRCN : All_other_revenues_not_classified_elsewhere
NPLGGREVSSOCCN : Social_insurance_contributions
NPLGGREVTAXTCN : Tax_revenues
NPLGGREVTOTHCN : All_other_taxes
NPLGGREVTOTLCN : Total_revenue_(and_grants)
NPLGGREVTRDECN : Taxes_on_international_trade
NPLGGREVTRDMCN : Customs_duty_on_imports
NPLGGREVTRDOCN : Other_taxes_on_international_trade
NPLGGREVTRDXCN : Customs_duty_on_exports
NPLGGREVTSHRCN : Revenue_Sharing_to_SNGs
```

Create an expenditures string

The fiscal scenarios below exogenize (hold constant) spending on those elements of government spending that are not being directly shocked.

To facilitate that, the variable `GGepx` is assigned a string containing all of the expenditure variables that are to beheld constant. This variable is then used when setting up each of the fiscal shocks below.

This list may need to be adjusted from model to model.

In [5]:
```
# Government spending variables to be held constant
GGexp=f'{Cty}GGEXPCAPTCN {Cty}GGEXPGNFSCN {Cty}GGEXPOTHRCN {Cty}GGEXPTOTHCN {Cty}GG




'''
NPLGGREVDCITER : NPLGGREVDCITER
NPLGGREVDPITER : NPLGGREVDPITER
NPLGGREVGEXCER : NPLGGREVGEXCER
NPLGGREVGVATER : NPLGGREVGVATER
NPLGGREVTRDMER : NPLGGREVTRDMER
NPLGGREVTRDXER : NPLGGREVTRDXER'''
```

Out[5]:
```
'\nNPLGGREVDCITER : NPLGGREVDCITER\nNPLGGREVDPITER : NPLGGREVDPITER\nNPLGGREVGEXCE
R : NPLGGREVGEXCER\nNPLGGREVGVATER : NPLGGREVGVATER\nNPLGGREVTRDMER : NPLGGREVTRDM
ER\nNPLGGREVTRDXER : NPLGGREVTRDXER'
```

The Indirect tax cut

This shock assumes that the main elements of government spending are held constant at their pre-shock levels. This assumption could be relaxed by commenting out the second line.

In the model indirect taxes are determined as a function of their lagged effective tax rate (Revenues divided by assumed tax base).

Inspecting the Goods and services tax revenue equation, one can note that the first term drops out in the forecast period (DUMH=0 in the forecast period) and the equation resolves into an identity where revenues evolve according to the tax base and the effective tax rate of the previous year.

To shock the level of indirect spending revenues in 2025 the add factor is used. By reducing revenues in 2025 by 1 percent of GDP, the effective tax rate in that year falls. For subsequent year's it is this lower effective rate that is retained, thereby making the shock permanent.

In [6]:
```
themodel['NPLGGREVD*CN'].df
```

Out[6]:

	NPLGGREVDCITCN	NPLGGREVDINVCN	NPLGGREVDPITCN	NPLGGREVDRCTCN
2023	118890.424652	44487.197174	79691.294606	243068.916432
2024	124786.858609	43155.411862	77173.114404	245115.384874
2025	135733.942701	44391.648752	79769.233703	259894.825156
2026	149859.070882	47249.452532	85311.291734	282419.815148
2027	167699.661107	51539.914492	92895.619837	312135.195435
2028	189823.581315	57203.951094	102192.134579	349219.666988
2029	216190.006575	64117.293373	113117.036853	393424.336801
2030	246366.499385	72105.822487	125667.729087	444140.050959

In [7]:

```
fpol_indirect=bline.copy()
fpol_indirect=themodel.fix(bline,f'{GGexp}') # Freeze other spending levels

# PIT and CIT tax revenues are determined by the lagged effective rate.
# Thus shocking the level of tax revenues in 2025 by one percent of GDP
# will increase the effective tax rate going forward by a constant amount
# consistent with a .5 percent increase in GDP and the tax base for consumption
fpol_indirect=fpol_indirect.mfcalc(
    f'<2025 2025> {Cty}GGREVGVATCN_A ={Cty}GGREVDCITCN_A - .01*{Cty}NYGDPMKTPCN')


#solve the model.
tempdf = themodel(fpol_indirect,silent=1,keep=f'1 % of GDP Indirect tax cut')
```

```
The folowing variables are fixed
NPLGGEXPCAPTCN
NPLGGEXPGNFSCN
NPLGGEXPOTHRCN
NPLGGEXPTOTHCN
NPLGGEXPTOTSCN
NPLGGEXPSUBSCN
NPLGGEXPWAGECN
```

DIrect tax hike of 1 % of GDP

The same basic methodology is followed for direct taxes.

In [8]:

```
fpol_direct=bline.copy()
fpol_direct=themodel.fix(bline,f'{GGexp}') # Freeze spending levels Capital

# In the model direct taxes (personal and corporate income taxes) tax are
# determined by the lagged effective rate of taxation.  Shocking the
# level of tax revenues in 2025 by one percent of GDP decreases the effective
# tax rate going forward by a constant amount consistent with a 1 percent decrease
# in GDP and the tax base for direct taxes (.5% for each)
fpol_direct=fpol_direct.mfcalc(
    f'<2025 2025> {Cty}GGREVDPITCN_A ={Cty}GGREVDPITCN_A  - .005*{Cty}NYGDPMKTPCN')
fpol_direct=fpol_direct.mfcalc(
```

```
    f'<2025 2025> {Cty}GGREVDCITCN_A ={Cty}GGREVDCITCN_A  - .005*{Cty}NYGDPMKTPCN')

#solve the model.
tempdf = themodel(fpol_direct,silent=1,keep=f'1 % of GDP direct tax cut')
```

```
The folowing variables are fixed
NPLGGEXPCAPTCN
NPLGGEXPGNFSCN
NPLGGEXPOTHRCN
NPLGGEXPTOTHCN
NPLGGEXPTOTSCN
NPLGGEXPSUBSCN
NPLGGEXPWAGECN
```

In [ ]:

```

```

Increase in expenditure on goods and services

The ex ante fiscal effort is the same in this scenario (1% of ex ante GDP) with the difference that it is implemented as an increase government spending, in this instance on goods and services.

In [9]:

```
fpol_ExpGS=bline.copy()
fpol_ExpGS=themodel.fix(bline,f'{GGexp}') # Freeze spending levels

fpol_ExpGS=fpol_ExpGS.mfcalc(
    f'<2025 2050> {Cty}GGEXPGNFSCN_X ={Cty}GGEXPGNFSCN_X   + .01*{Cty}NYGDPMKTPCN')
```

```
The folowing variables are fixed
NPLGGEXPCAPTCN
NPLGGEXPGNFSCN
NPLGGEXPOTHRCN
NPLGGEXPTOTHCN
NPLGGEXPTOTSCN
NPLGGEXPSUBSCN
NPLGGEXPWAGECN
```

In [10]:

```
#solve the model.
tempdf = themodel(fpol_ExpGS,silent=1,keep=f'1 % of GDP increase in G&S spending')
#themodel.lastdf['IDNNECONGOVTCN']/themodel.basedf['IDNNECONGOVTCN']
```

Increase in expenditure on investment goods

The ex ante fiscal effort is the same in this scenario (1% of ex ante GDP), implemented as an increase in government spending on capital goods.

In [11]:

```
fpol_ExpInv=bline.copy()
fpol_ExpInv=themodel.fix(bline,f'{GGexp}') # Freeze spending levels

fpol_ExpInv=fpol_ExpInv.mfcalc(f'<2025 2050> {Cty}GGEXPCAPTCN_X ={Cty}GGEXPCAPTCN_X
```

```
The folowing variables are fixed
NPLGGEXPCAPTCN
NPLGGEXPGNFSCN
NPLGGEXPOTHRCN
NPLGGEXPTOTHCN
NPLGGEXPTOTSCN
NPLGGEXPSUBSCN
NPLGGEXPWAGECN
```

In [12]:
```
#solve the model.
tempdf = themodel(fpol_ExpInv,silent=1,keep=f'1 % of GDP increase in Govt investmen
#themodel.lastdf['IDNNECONGOVTCN']/themodel.basedf['IDNNECONGOVTCN']
```

In [ ]:

Increase in expenditure on Transfers to households

In this scenario the same fiscal effort is implemented as an increase in transfers to households.

In [13]:
```
fpol_ExpTrans=bline.copy()
fpol_ExpTrans=themodel.fix(bline,f'{GGexp}') # Freeze spending levels

fpol_ExpTrans=fpol_ExpTrans.mfcalc(f'<2025 2050> {Cty}GGEXPTSOCCN_X ={Cty}GGEXPTSOC
```

```
The folowing variables are fixed
NPLGGEXPCAPTCN
NPLGGEXPGNFSCN
NPLGGEXPOTHRCN
NPLGGEXPTOTHCN
NPLGGEXPTOTSCN
NPLGGEXPSUBSCN
NPLGGEXPWAGECN
```

In [14]:
```
#solve the model.
tempdf = themodel(fpol_ExpTrans,silent=1,keep=f'1 % of GDP increase in transfers to
#themodel.lastdf['IDNNECONGOVTCN']/themodel.basedf['IDNNECONGOVTCN']
```

Comparisons of results from the fiscal scenarios

The following charts compare results from the different fiscal simulations. Impacts will differ both in terms of their long-term and short-term impacts. For example a scenario that increased investment would likely have negative impacts on consumption in the short-run but in the longer run could be expected to have an opposite impact on potential output, GDP and perhaps consumption. A scenario that concentrated on transfers or consumption might have more of a short-term impact on demand but in the long run would have limited (and potentially negative impacts on output), especially if increased fiscal deficits and debt crowded out private sector investment. As all World Bank models are customized to the country for which they have been built the extent of these effects can vary across models.

Recall the scenarios that were run by interrogating the keep_solutions dictionary.

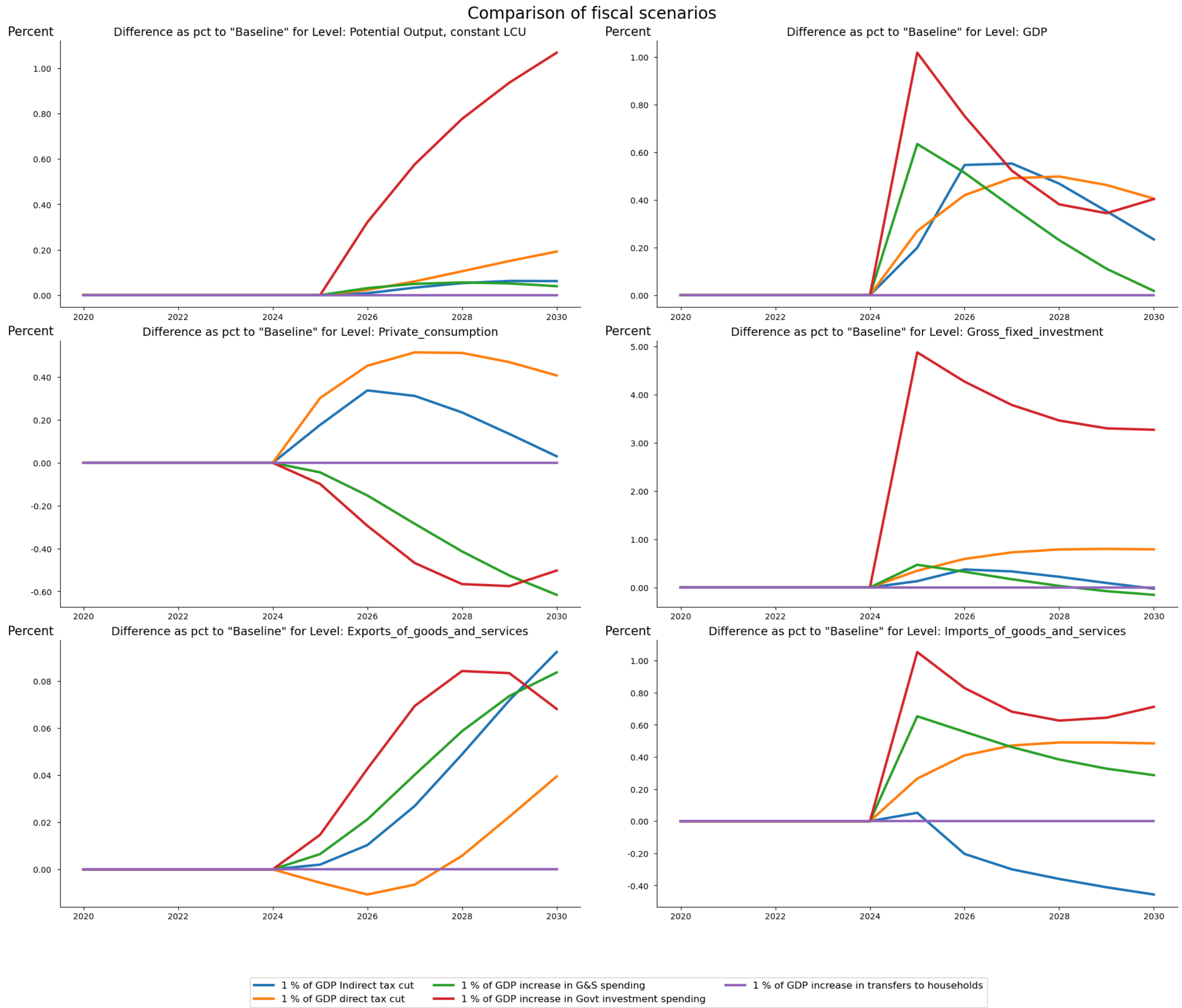
In [15]:
```
for key,value in themodel.keep_solutions.items():
    print(key)
```

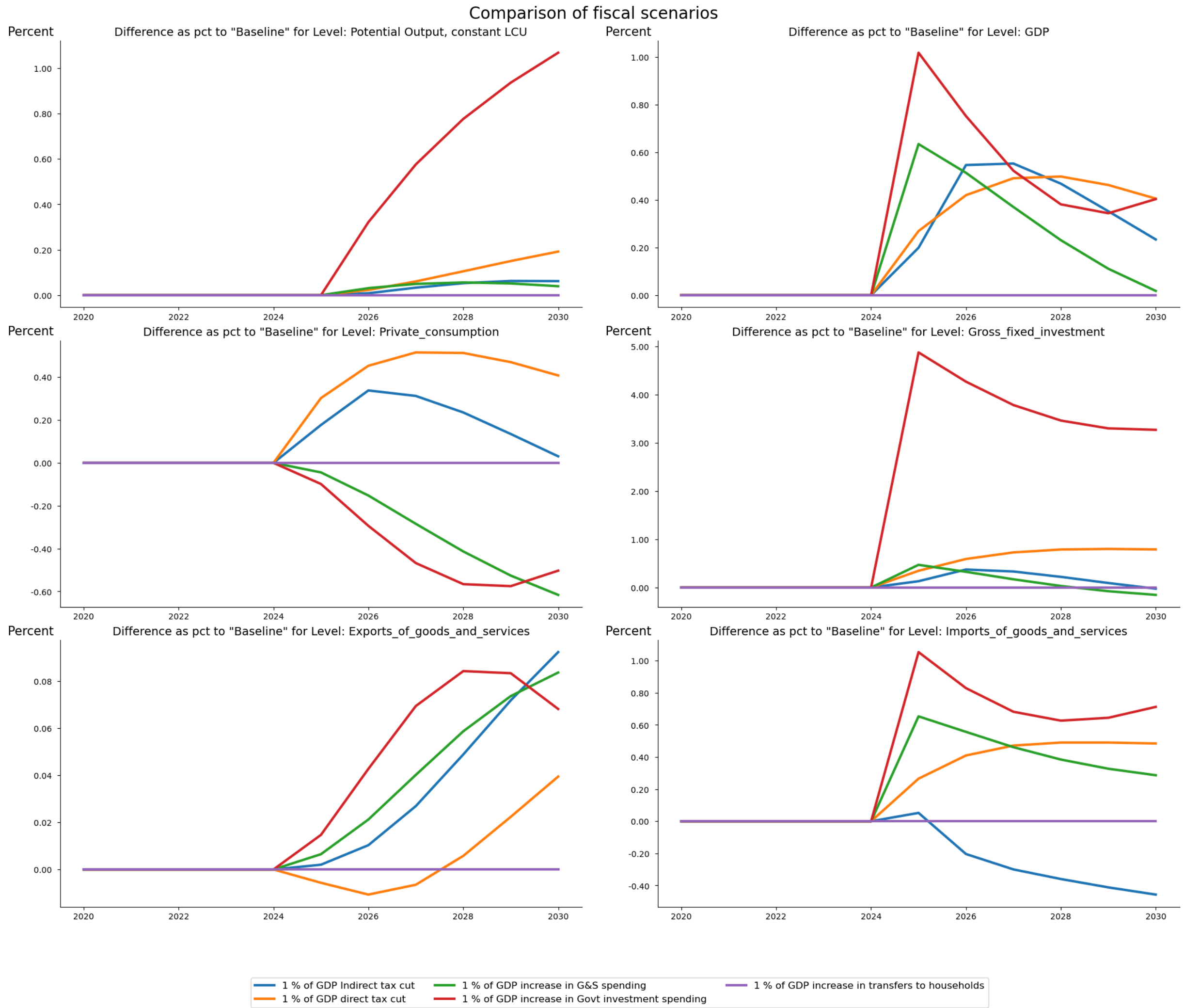
```
Baseline
1 % of GDP Indirect tax cut
1 % of GDP direct tax cut
1 % of GDP increase in G&S spending
1 % of GDP increase in Govt investment spending
1 % of GDP increase in transfers to households
```

Real GDP impacts and impacts on main Real GDP expenditure components

The following chart use the `with keep_switch()` clause to select which scenario results are to be plotted, and then plots them on a series of charts using the `.keep_plot()` method with the `samefig` and `diffpct` options set to True. the first option ensures that all the graphs are arranged in a grid and the second expresses the results as a percent deviation from the results in the first scenario specified in the `keep_switch()` -- in this case the baseline scenario. Note, the included scenarios in the `keep_switch()` command are identified by the text used in the initial keep command and separated by a horizontal line "|".

In [16]:
```
with themodel.keepswitch(scenarios='Baseline|1 % of GDP Indirect tax cut|1 % of GDP
    themodel.keep_plot(f'{Cty}NYGDPPOTLKN {Cty}NYGDPMKTPKN {Cty}NECONPRVTKN {Cty}NE
                       start=2020, end=2040,diffpct=True,samefig=True,legend=True,t
```


Comparison of fiscal scenarios
Difference as pct to "Baseline" for Level: Potential Output, constant LCU
Difference as pct to "Baseline" for Level: GDP
Difference as pct to "Baseline" for Level: Private_consumption
Difference as pct to "Baseline" for Level: Gross_fixed_investment
Difference as pct to "Baseline" for Level: Exports_of_goods_and_services
Difference as pct to "Baseline" for Level: Imports_of_goods_and_services
Percent
1 % of GDP Indirect tax cut
1 % of GDP direct tax cut
1 % of GDP increase in G&S spending
1 % of GDP increase in Govt investment spending
1 % of GDP increase in transfers to households


Comparison of fiscal scenarios
Percent
Difference as pct to "Baseline" for Level: Potential Output, constant LCU
Percent
Difference as pct to "Baseline" for Level: GDP
Percent
Difference as pct to "Baseline" for Level: Private_consumption
Percent
Difference as pct to "Baseline" for Level: Gross_fixed_investment
Percent
Difference as pct to "Baseline" for Level: Exports_of_goods_and_services
Percent
Difference as pct to "Baseline" for Level: Imports_of_goods_and_services
1 % of GDP Indirect tax cut
1 % of GDP direct tax cut
1 % of GDP increase in G&S spending
1 % of GDP increase in Govt investment spending
1 % of GDP increase in transfers to households

For Nepal, the GDP results appear to be consistent with expectations. Fiscal expansion of all types boosted demand and GDP in the short run. However, the long run the impact depends on the impact of the spending on potential output. Sustained increased investment spending served to increase the capital stock and contribute to higher potential and actual GDP. Spending that focused on consumption or transfers has little discernible impact on potential, with the principal engine -- with a short-term increase in investment spending contributing to higher potential but this effect dieing out toward the end of the projection period.

Transfers to households and decreased direct taxation tended to benefit households, although this impact declined over time due to the relative stability of potential output in these scenarios. The cut in indirect taxes tended to generate small but persistent benefits, presumably due to lower prices for domestically produced goods with benefits to both exports and import competing goods.

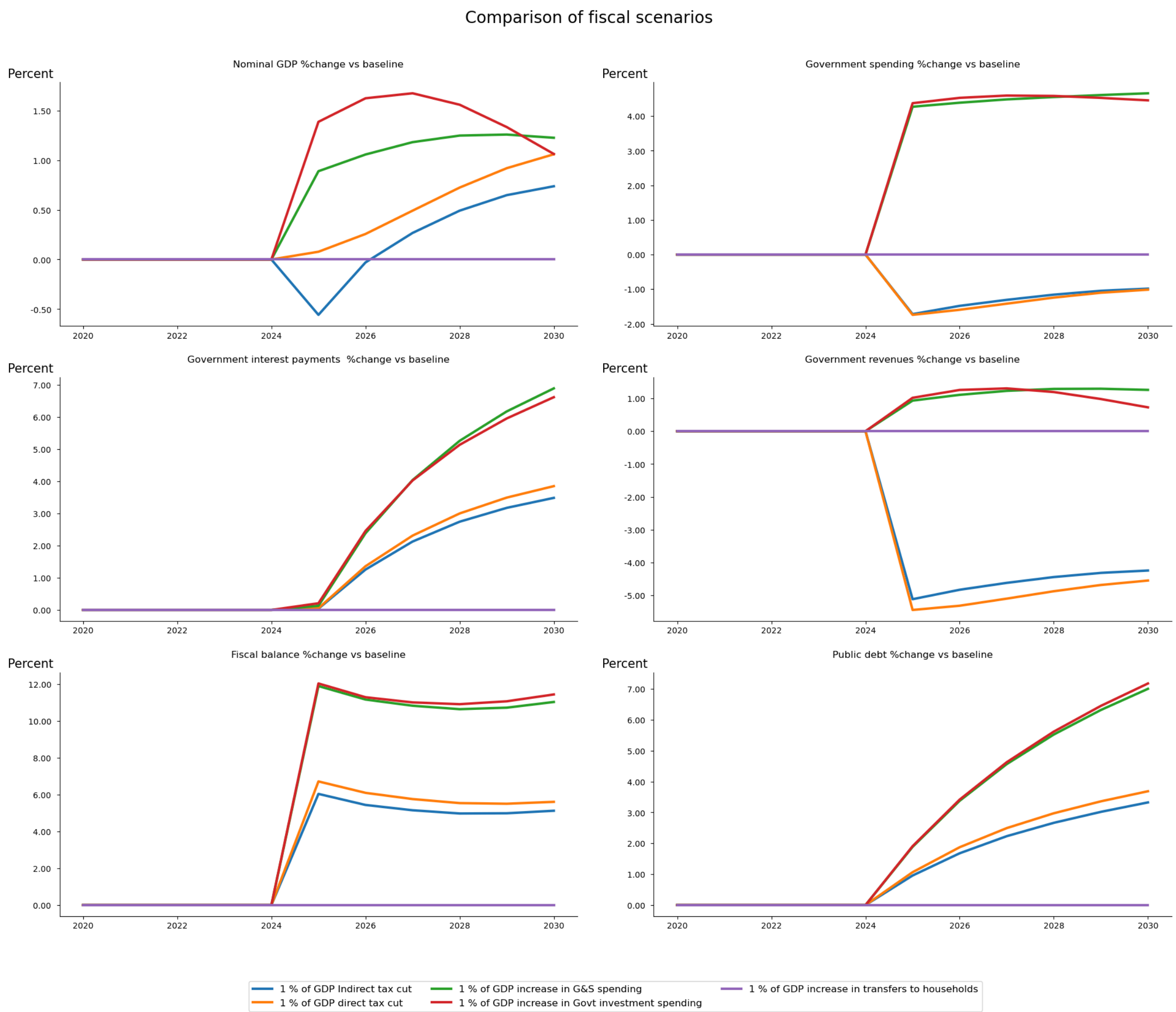
Impacts on the fiscal accounts

The following command shows the impacts on several of the main fiscal indicators.

In [17]:
```
import matplotlib.pyplot as plt
themodel.keep_solutions.keys()
```

Out[17]:
```
dict_keys(['Baseline', '1 % of GDP Indirect tax cut', '1 % of GDP direct tax cut',
'1 % of GDP increase in G&S spending', '1 % of GDP increase in Govt investment spe
nding', '1 % of GDP increase in transfers to households'])
```

In [18]:
```
with themodel.keepswitch(
    scenarios='''Baseline|1 % of GDP Indirect tax cut|1 % of GDP direct tax cut|1 %
    all_figs=themodel.keep_plot(
        f'{Cty}NYGDPMKTPCN {Cty}GGEXPTOTLCN {Cty}GGEXPINTPCN {Cty}GGREVTOTLCN {Cty}
        start=2020, end=2040,diffpct=True,samefig=True,legend=True,title="Compariso

#The combined figures are called onefig
figs = all_figs['onefig']
figs.axes[0].set_title("\nNominal GDP %change vs baseline\n")
figs.axes[1].set_title("\nGovernment spending %change vs baseline\n")
figs.axes[2].set_title("\nGovernment interest payments  %change vs baseline\n")
figs.axes[3].set_title("\nGovernment revenues %change vs baseline\n")
figs.axes[4].set_title("\nFiscal balance %change vs baseline\n")
figs.axes[5].set_title("\nPublic debt %change vs baseline\n")
plt.show()
```



Note that:

- Nominal GDP is increased in all scenarios, mainly reflecting the inflationary impact of the scenario (most scenarios saw real GDP decline).
- The fiscal account deteriorates in all scenarios as compared with the baseline.
    - Nominal spending increases even in the tax scenarios, but here the driver is increased interest payments as other elements of spending were held constant.
    - Although interest payments as a percent of their initial level are up a lot, the increase as a percent of GDP (see next set of charts is less pronounced).
    - Revenues improve in the spending scenarios because of higher nominal GDP.
    - The fiscal balance deteriorates (becomes more negative) with the extent of the deterioration smallest in the scenarios where real GDP growth is increasing of hit less hard.
    - Public debt is higher in all scenarios

Fiscal impacts as a percent of GDP

As observed. higher inflation (due to increased demand in the early years of the simulation) mean that both revenues and expenditures are higher in the simulation scenarios.

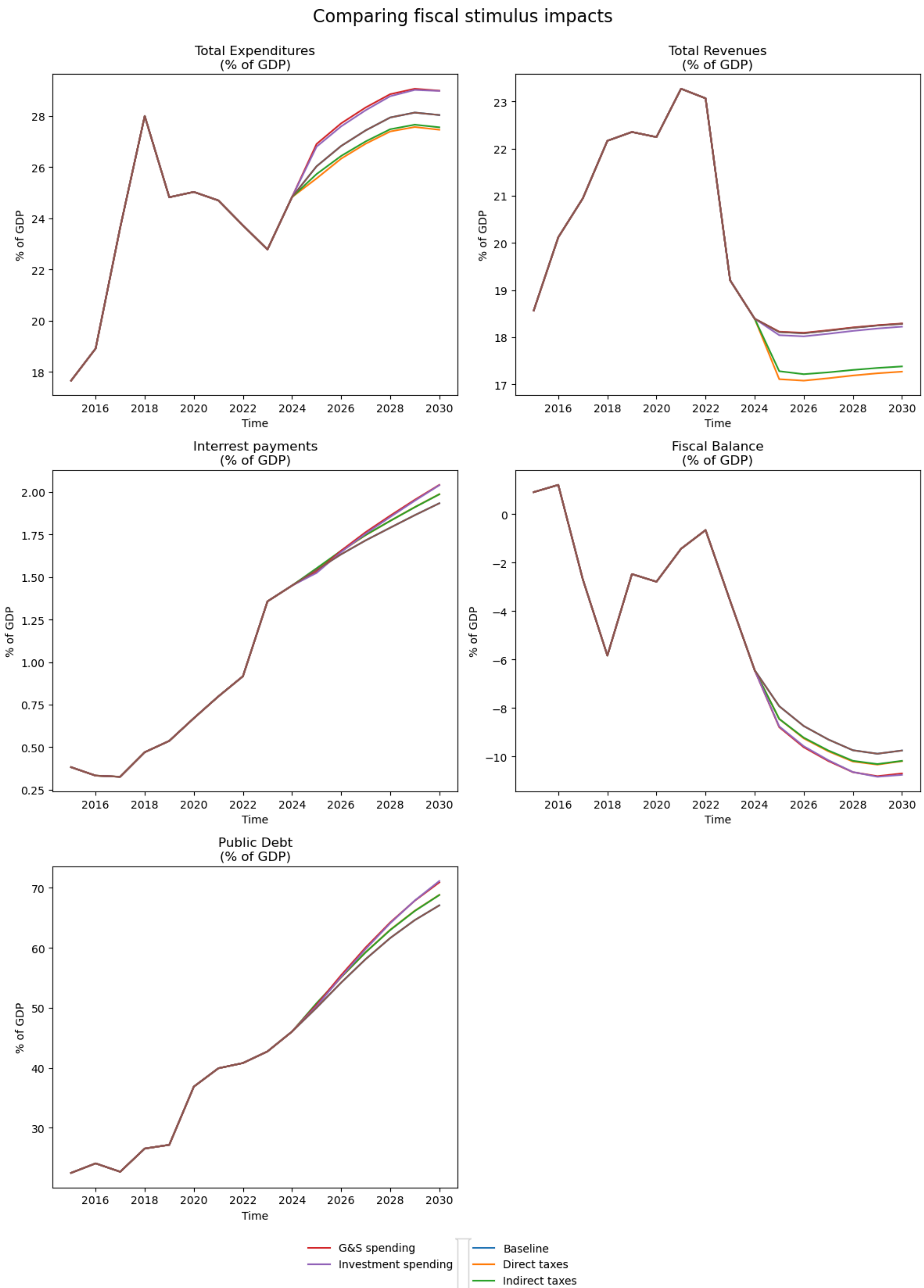
To correct for this effect, the following charts show the results as a percent of GDP. Here the inflation influences both the numerator and the denominator, so just the net effect is drawn.

In [19]:
```
# Uncomment to display the mnemonics and desriptions of the general gobernment (GG)
# (EXP) in nominal local currency (CN)
#themodel[f'{Cty}GGEXP*CN'].des
```

In [20]:
```
for key,df in themodel.keep_solutions.items():
    df[f'{Cty}GGBALOVRLCN_']=df[f'{Cty}GGBALOVRLCN']/df[f'{Cty}NYGDPMKTPCN']*100
    df[f'{Cty}GGEXPTOTLCN_']=df[f'{Cty}GGEXPTOTLCN']/df[f'{Cty}NYGDPMKTPCN']*100
    df[f'{Cty}GGREVTOTLCN_']=df[f'{Cty}GGREVTOTLCN']/df[f'{Cty}NYGDPMKTPCN']*100
    df[f'{Cty}GGEXPINTPCN_']=df[f'{Cty}GGEXPINTPCN']/df[f'{Cty}NYGDPMKTPCN']*100
    df[f'{Cty}GGDBTTOTLCN_']=df[f'{Cty}GGDBTTOTLCN']/df[f'{Cty}NYGDPMKTPCN']*100
    themodel.keep_solutions[key]=df
```

In [21]:
```
base = themodel.keep_solutions['Baseline'].loc[2015:]
InDirect = themodel.keep_solutions['1 % of GDP Indirect tax cut'].loc[2015:]
GoodsServices = themodel.keep_solutions['1 % of GDP increase in G&S spending'].loc[
Invest = themodel.keep_solutions['1 % of GDP increase in Govt investment spending']
Transfers = themodel.keep_solutions['1 % of GDP increase in transfers to households
Direct = themodel.keep_solutions['1 % of GDP direct tax cut'].loc[2015:]


figcombo,axs=plt.subplots(3,2,figsize=(12,16))
i=0

vars=['GGEXPTOTLCN','GGREVTOTLCN','GGEXPINTPCN','GGBALOVRLCN','GGDBTTOTLCN']
for idx,var in enumerate(vars):
    row = idx // 2  # Calculate the row index based on the two-column grid
    col = idx % 2
    axs[row,col].plot(base[f'{Cty}{var}_'],label='Baseline')
    axs[row,col].plot(Direct[f'{Cty}{var}_'],label='Direct taxes')
    axs[row,col].plot(InDirect[f'{Cty}{var}_'],label='Indirect taxes')
    axs[row,col].plot(GoodsServices[f'{Cty}{var}_'],label='G&S spending')
    axs[row,col].plot(Invest[f'{Cty}{var}_'],label='Investment spending')
    axs[row,col].plot(Transfers[f'{Cty}{var}_'],label='Transfers')
    axs[row,col].set_ylabel('% of GDP')
    axs[row,col].set_xlabel('Time')

axs[0,0].set_title('Total Expenditures\n(% of GDP)')
axs[0,1].set_title('Total Revenues\n(% of GDP)')
axs[1,0].set_title('Interrest payments\n(% of GDP)')
axs[1,1].set_title('Fiscal Balance\n(% of GDP)')
axs[2,0].set_title('Public Debt\n(% of GDP)')

#delete the blank chart
figcombo.delaxes(axs[2, 1])

# Add a joint legend below the subplots with only the first two elements
handles, labels = axs[0,0].get_legend_handles_labels()
```

```
#Use only the labels of the first six series in the legend
figcombo.legend(handles[:5], labels[:5], loc='upper center', bbox_to_anchor=(0.5, 0
figcombo.suptitle("Comparing fiscal stimulus impacts\n", fontsize=16)
# Adjust layout to prevent overlapping titles and labels
plt.tight_layout()
plt.show()
```



- Spending as a percent of GDP increases by as much as 2 percent of GDP, the result of the original bump up in spending plus increased interest payments as the debt rises.

- Revenues as a percent of GDP are down in the tax reduction scenarios, but up somewhat in the spending scenarios -- presumably reflecting a switch in the mix of total expenditure towards categories that are relatively highly taxed.
- Interest payments rise by as much as 3 percent of GDP by the end of the period due to higher debt levels and because of higher interest rates as debt to GDP rates rise.
- Higher debt and fiscal borrowing will translate into increased competition for domestic and foreign savings, crowding out private sector investment
- The fiscal balance deteriorates by one and three percent of GDP, with differences reflecting differences in real GDP and inflation impacts, and revenue impacts.
- Debt rises by more than 30 percent of GDP as the permanent 1 percent increase in spending accumulates over time.

Non fiscal simulations

Three non-fiscal scenarios were run. The first a temporary increase in the monetary policy interest rate, the second a 10 percent depreciation and the final a permanent $20 increase in the price of crude_petrol.

Monetary policy shock

In this shock, it is assumed that the central bank raises its policy rate by 1 percentage point for 1 year.

In [64]:
```
Mpol=bline.copy()
Mpol=themodel.fix(bline,f'{GGexp}') # Freeze spending levels

Mpol=themodel.fix(Mpol,f'{Cty}FMLBLPOLYFR',2025,2025) # One year shock to MP but th
Mpol=Mpol.mfcalc(f'<2025 2025> {Cty}FMLBLPOLYFR_X ={Cty}FMLBLPOLYFR_X   + 1')
#Mpol[f'{Cty}FMLBLPOLYFR_X'].loc[2020:2030]/Mpol[f'{Cty}FMLBLPOLYFR'].loc[2020:2030
```

```
The folowing variables are fixed
NPLGGEXPCAPTCN
NPLGGEXPGNFSCN
NPLGGEXPOTHRCN
NPLGGEXPTOTHCN
NPLGGEXPTOTSCN
NPLGGEXPSUBSCN
NPLGGEXPWAGECN
The folowing variables are fixed
NPLFMLBLPOLYFR
```

In [69]:
```
#solve the model.
tempdf = themodel(Mpol,silent=1,keep=f'1 ppt increase in policy rate in 2025')
#themodel.lastdf['IDNNECONGOVTCN']/themodel.basedf['IDNNECONGOVTCN']

themodel[f'{Cty}FMLBLPOLYFR'].frml
```

```
NPLFMLBLPOLYFR : FRML <DAMP,STOC> NPLFMLBLPOLYFR = (100*NPLFMLBLPOLYFR_A+100* (0.696
375694848535*NPLFMLBLPOLYFR(-1)/100+0.28808154468723*INDFMLBLPOLYFR/100+(1-0.6963756
94848535-0.28808154468723)*(NPLNYGDPGAP_/100)-0.0166426247204525*DURING_2017) ) * (1
-NPLFMLBLPOLYFR_D)+ NPLFMLBLPOLYFR_X*NPLFMLBLPOLYFR_D $
```

In [71]:
```python
themodel.lastdf[f'{Cty}FMLBLPOLYFR'].loc[2020:2030]-themodel.basedf[f'{Cty}FMLBLPOL
```

Out[71]:
```
2020    0.000000e+00
2021    0.000000e+00
2022    0.000000e+00
2023    0.000000e+00
2024   -8.704149e-14
2025    1.000000e+00
2026    6.868818e-01
2027    4.697575e-01
2028    3.219558e-01
2029    2.233120e-01
2030    1.584902e-01
Name: NPLFMLBLPOLYFR, dtype: float64
```

Exchange rate depreciation

This shock assumes a depreciation of the currency by 10 percent in 2025.

In [53]:
```python
Mpol_exr=bline.copy()
#Mpol_exr=themodel.fix(bline,f'{GGexp}') # Freeze spending levels

Mpol_exr=themodel.fix(Mpol,f'{Cty}PANUSATLS',2025,2050) # One year shock to MP but
Mpol_exr=Mpol_exr.mfcalc(f'<2025 2050> {Cty}PANUSATLS_X ={Cty}PANUSATLS_X * 1.1')
```

```
The folowing variables are fixed
NPLPANUSATLS
```

In [54]:
```python
#solve the model.
tempdf = themodel(Mpol_exr,silent=1,keep=f'Permanent 10 percent depreciation in 202
```

TFP Shock

This shock explores the effect of a permanent increase in the level of TFP by 1 percent beginning in 2025.

In [55]:
```python
TFP=bline.copy()
TFP=themodel.fix(bline,f'{GGexp}') # Freeze spending levels

#TFP=themodel.fix(Mpol,f'{Cty}PANUSATLS',2025,2050) # One year shock to MP but then
TFP=TFP.mfcalc(f'<2025 2050> {Cty}NYGDPTFP ={Cty}NYGDPTFP * 1.01')
```

```
The folowing variables are fixed
NPLGGEXPCAPTCN
NPLGGEXPGNFSCN
NPLGGEXPOTHRCN
NPLGGEXPTOTHCN
NPLGGEXPTOTSCN
NPLGGEXPSUBSCN
NPLGGEXPWAGECN
```

In [56]:
```
#solve the model.
tempdf = themodel(TFP,silent=1,keep=f'A permanent 1 percent increase in TFP levels'
#themodel.lastdf['IDNNECONGOVTCN']/themodel.basedf['IDNNECONGOVTCN']
```

A permanent 20 percent increase in oil prices

This shock explores the sensitivity of the model to a permanent $20 increase in global oil and natural gas prices beginning in 2025. The natural gas price are as$
20 increase in world crude oil prices.

In [57]:
```
themodel['WLDF*GAS*'].des
```

```
WLDFNGAS_EUR             : Price of natural gas (USD)
WLDFNGAS_EUR_VALUE_2015 : WLDFNGAS_EUR_VALUE_2015
```

In [58]:
```
Oil=bline.copy()
Oil=themodel.fix(bline,f'{GGexp}') # Freeze spending levels

# scenario to raise oil prices.  Assume that natural gas prices increase
# by a similar margin as a $20 increase in crude oil prices
Oil=Oil.mfcalc(
    f'<2025 2050> WLDFNHAS_EUR = WLDFNHAS_EUR * (WLDFCRUDE_PETRO+20)/WLDFCRUDE_PETR
Oil=Oil.mfcalc(
    f'<2025 2050> WLDFCRUDE_PETRO = WLDFCRUDE_PETRO + 20')
```

```
The folowing variables are fixed
NPLGGEXPCAPTCN
NPLGGEXPGNFSCN
NPLGGEXPOTHRCN
NPLGGEXPTOTHCN
NPLGGEXPTOTSCN
NPLGGEXPSUBSCN
NPLGGEXPWAGECN
```

In [59]:
```
#solve the model.
tempdf = themodel(Oil,silent=1,keep=f'A permanent $20 increase in global crude oil
#themodel.lastdf['IDNNECONGOVTCN']/themodel.basedf['IDNNECONGOVTCN']
```

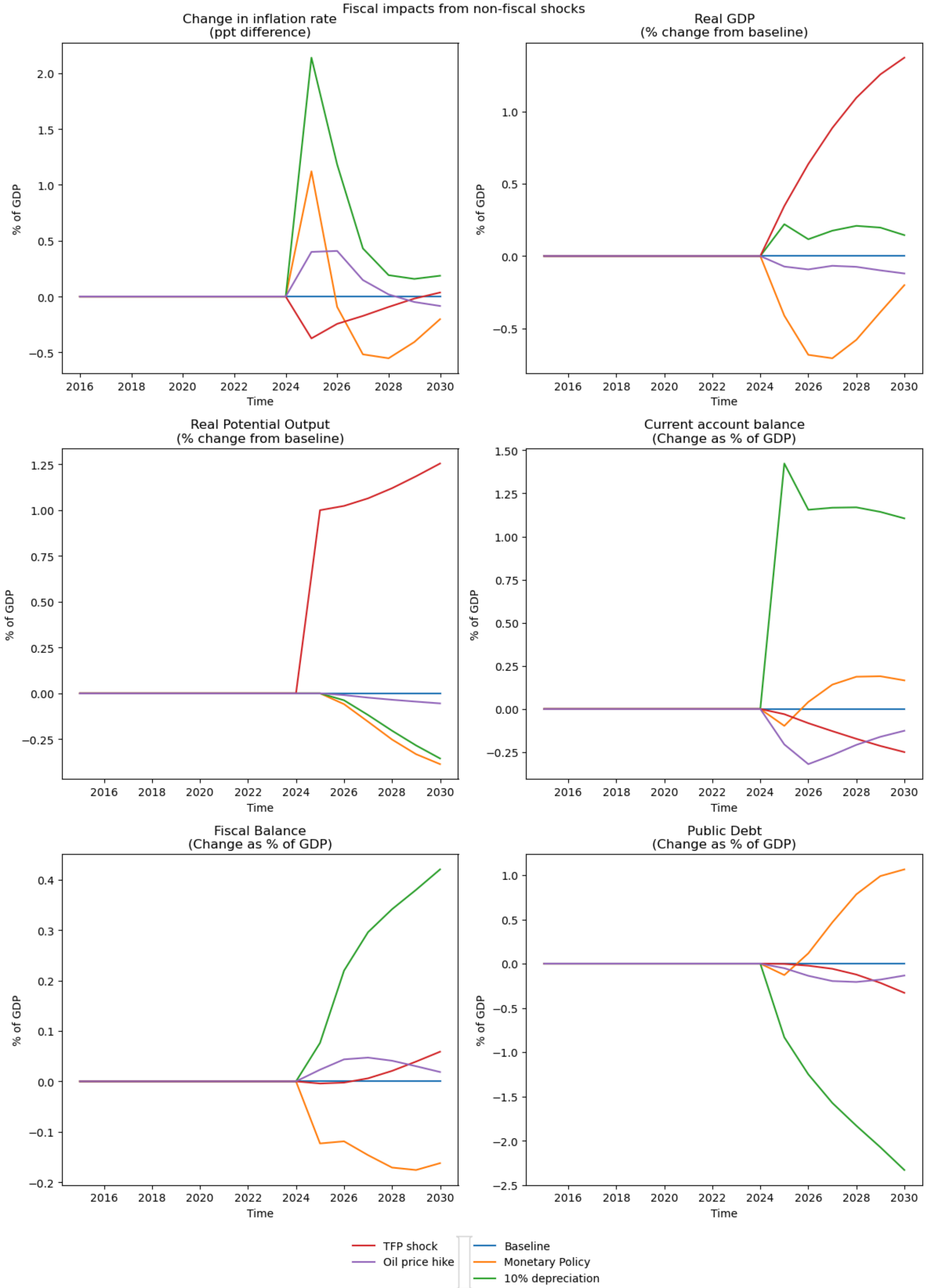
Summary impacts of non-fiscal scenarios

The following graphs show the change in the main macroeconomic indicators (Inflation, proxied here by the inflation rate of household consumption), real GDP, and changes in the current account, fiscal balances and debt levels expressed as a percent of GDP.

In [60]:

```
for key,df in themodel.keep_solutions.items():
    df[f'{Cty}NECONPRVTXN_COMP']=(df[f'{Cty}NECONPRVTXN'].pct_change()-base[f'{Cty}
    df[f'{Cty}NYGDPMKTPKN_COMP']=(df[f'{Cty}NYGDPMKTPKN']/base[f'{Cty}NYGDPMKTPKN']
    df[f'{Cty}NYGDPPOTLKN_COMP']=(df[f'{Cty}NYGDPPOTLKN']/base[f'{Cty}NYGDPPOTLKN']
    df[f'{Cty}BNCABFUNDCD_COMP']=((df[f'{Cty}BNCABFUNDCD']/df[f'{Cty}NYGDPMKTPCD'])
    df[f'{Cty}GGBALOVRLCN_COMP']=((df[f'{Cty}GGBALOVRLCN']/df[f'{Cty}NYGDPMKTPCN'])
    df[f'{Cty}GGDBTTOTLCN_COMP']=((df[f'{Cty}GGDBTTOTLCN']/df[f'{Cty}NYGDPMKTPCN'])
    themodel.keep_solutions[key]=df

base = themodel.keep_solutions['Baseline'].loc[2015:]
MP = themodel.keep_solutions['1 ppt increase in policy rate in 2025'].loc[2015:]
EXR = themodel.keep_solutions['Permanent 10 percent depreciation in 2025'].loc[2015
TFP = themodel.keep_solutions['A permanent 1 percent increase in TFP levels'].loc[2
Oil = themodel.keep_solutions['A permanent 20 increase in global crude-oil prices']

figcombo1,axs=plt.subplots(3,2,figsize=(12,16))
i=0

vars=['NECONPRVTXN','NYGDPMKTPKN','NYGDPPOTLKN','BNCABFUNDCD','GGBALOVRLCN','GGDBTT
for idx,var in enumerate(vars):
    row = idx // 2  # Calculate the row index based on the two-column grid
    col = idx % 2
    axs[row,col].plot(base[f'{Cty}{var}_COMP'],label='Baseline')
    axs[row,col].plot(MP[f'{Cty}{var}_COMP'],label='Monetary Policy')
    axs[row,col].plot(EXR[f'{Cty}{var}_COMP'],label='10% depreciation')
    axs[row,col].plot(TFP[f'{Cty}{var}_COMP'],label='TFP shock')
    axs[row,col].plot(Oil[f'{Cty}{var}_COMP'],label='Oil price hike')
    axs[row,col].set_ylabel('% of GDP')
    axs[row,col].set_xlabel('Time')

axs[0,0].set_title('Change in inflation rate\n(ppt difference)')
axs[0,1].set_title('Real GDP\n(% change from baseline)')
axs[1,0].set_title('Real Potential Output\n(% change from baseline)')
axs[1,1].set_title('Current account balance\n(Change as % of GDP)')
axs[2,0].set_title('Fiscal Balance\n(Change as % of GDP)')
axs[2,1].set_title('Public Debt\n(Change as % of GDP)')


# Add a joint legend below the subplots with only the first two elements
handles, labels = axs[0,0].get_legend_handles_labels()

#Use only the labels of the first six series in the legend
figcombo1.legend(handles[:6], labels[:6], loc='upper center', bbox_to_anchor=(0.5,
figcombo1.suptitle("Fiscal impacts from non-fiscal shocks")
# Adjust layout to prevent overlapping titles and labels
```

```
plt.tight_layout()
plt.show()
```



In [46]:
```
themodel['IDNNYGDP*KN'].des
```

```
Error: The variable specification:"IDNNYGDP*KN" did not generate any matches
```

Out[46]:
```
<function modelvis.DummyVis.__getattr__.<locals>.dummy_method(*args, **kwargs)>
```

As might be expected, effects across these scenarios are more divergent.

Inflation Higher oil prices and an exchange rate depreciation both are inflationary at least in the short-run. The inflation impact in the oil scenario is much smaller than in the depreciation scenario and dissipates quite rapidly as the economy adjusts. In contrast , **there appears to be a permanent increase in inflation in the exchange rate scenario -- odd.**

Both the TFP shock and the monetary policy tightening contribute to declines in inflation. The TFP shock is deflationary because it raises potential output and therefore open up a negative output gap, at least initially. Over time this effect diminished but does not seem to go away **(odd)**. In contrast the monetary policy shock which also induces an increase in the output gap (due to lower demand) in the short run, tends to recover its pre-shock inflation rate.

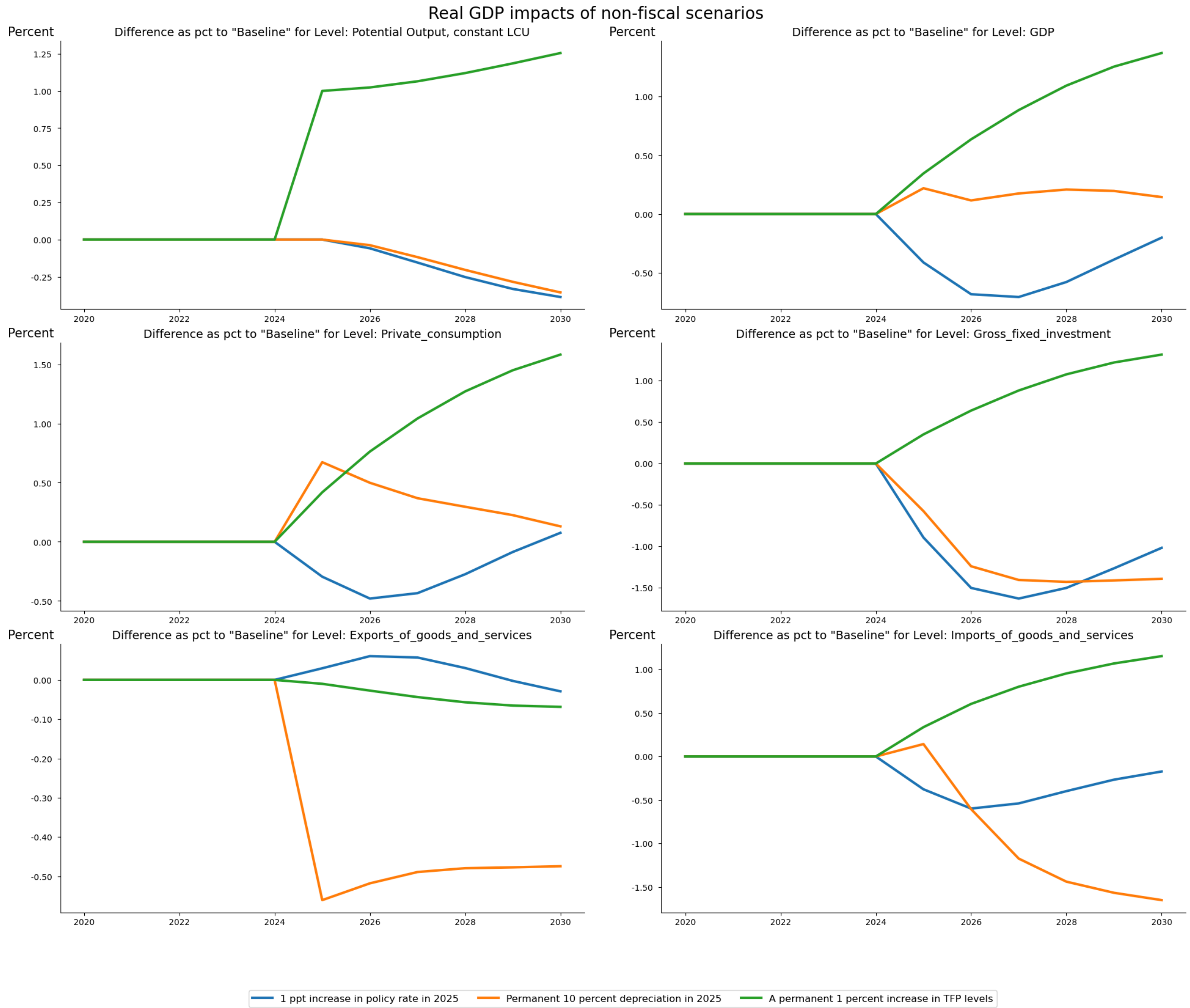
There is something wrong in the inflation response,. Inflation should return to its pre-shock level but it really is not doing so in any of the scenarios.

The permanent increase in TFP translates fairly quickly into higher GDP, and GDP continues to rise -- reflecting stronger potential output. While the initial increase in TFP will have boosted potential the subsequent increase reflects higher investment induced by the increased productivity of capital (and labor).

The potential impact is too large (triple the initial TFP shock) should be more like 1.3-1.5 or so. Whats happening to labor? Is this a non-oil GDP effect

Real Gdp impacts of non-fiscal simulations

In [47]:
```
with themodel.keepswitch(scenarios='Baseline|1 ppt increase in policy rate in 2025|
    themodel.keep_plot(f'{Cty}NYGDPPOTLKN {Cty}NYGDPMKTPKN {Cty}NECONPRVTKN {Cty}NE
                           start=2020, end=2050,diffpct=True,samefig=True,legend=True,t
```



Discussion of real GDP impacts of non-fiscal scenarios

The TFP shock (green line)

Potential and real GDP impacts mirror one another, with the TFP shock raising potential output permanently, and with real GDP catching up over-time slowly. The potential GDP impact rises proportionately over time because the higher output induces additional investment which adds further to potential GDP. Higher potential and actual GDP translates into increased consumption, exports and imports as the economy adjusts to the higher scale of activity.

The monetary policy shock (blue line)

As can be expected a tightening of monetary policy has negative effects on GDP. Higher interest rates and slower growth reduce investment growth which has a modest negative effect on potential output and long-run GDP. Consumer demand mirrors GDP as incomes are reduced modestly as compared with the baseline. Exports are hurt initially due to high capital costs, but as inflation declines they benefit and by the end of the period the impact is negligible. Imports are similarly hurt initially but recover most of the losses, They remain lower than baseline in line with lower domestic demand and GDP.

Depreciation (orange line)

The long run effect of the 10 percent depreciation is negative, reflecting higher import costs, which dampen investment and contribute to a cumulative 1.5 percent decline in potential output. Exports benefit initially as domestic goods become more compEtitive abroad and imports decline as import-competing firms benefit. However, higher import costs translate into lower real incomes and lower consumption, which persist into the future due to the investment effect on GDP.

Oil price hike (red line)

A permanent $20 nominal increase in oil prices has modest impacts on GDP, increasing potential slightly in Indonesia an oil exporter as incomes rise allowing for a modest increase in investment and consumption. Real non-oil exports decline, mainly because of price effects induced by the additional spending that the higher oil revenues enable, which are in turn reflected in a modest decrease in import volumes.

Fiscal impacts of non-fiscal scenarios

The depreciation fall in spending is kind of hard to explain.

In [48]:

```
# Uncomment to display the mnemonics and desriptions of the general gobernment (GG)
# (EXP) in nominal local currency (CN)
themodel[f'{Cty}GGEXP*CN'].des
```

```
NPLGGEXPCAPTCN : Capital_Expenditures
NPLGGEXPCRNTCN : Current_expenditures
NPLGGEXPGCAPCN : Social_Service_Capital_Grants_Expenditure
NPLGGEXPGCURCN : Social_Service_Recurrent_Grants_Expenditure
NPLGGEXPGGCACN : Grant_to_Other_Levels_of_Government_Capital
NPLGGEXPGGFTCN : Grant_to_Other_Levels_of_Government_Fiscal_Transfers
NPLGGEXPGGOTCN : Grant_to_Other_Levels_of_Government_Others
NPLGGEXPGGOVCN : Grant_to_Other_Levels_of_Government
NPLGGEXPGITLCN : Grants_Expenditure_Intl_Org_Membership
NPLGGEXPGNFSCN : Goods_and_services
NPLGGEXPGRNTCN : Grants_Expenditure
NPLGGEXPGSSVCN : Social_Service_Grants_Expenditure
NPLGGEXPINTDCN : Domestic_debt_interest_payments
NPLGGEXPINTECN : External_debt_interest_payments
NPLGGEXPINTPCN : Interest_payments
NPLGGEXPOTHRCN : All_other_Expenditures_not_classified_elsewhere
NPLGGEXPSUBSCN : Subsidies
NPLGGEXPTOTHCN : Other_Current_Expenditure
NPLGGEXPTOTLCN : Total_expenditure
NPLGGEXPTOTSCN : Social_Securities
NPLGGEXPTPNSCN : Pension_Employment_Related_to_Social_Benefit_of_Employees
NPLGGEXPTRNSCN : Current_transfers
NPLGGEXPTSOCCN : Social_Assistance
NPLGGEXPTSSCCN : Total_Social_Securities_Expenditure
NPLGGEXPWAGECN : Compensation_of_employees
```

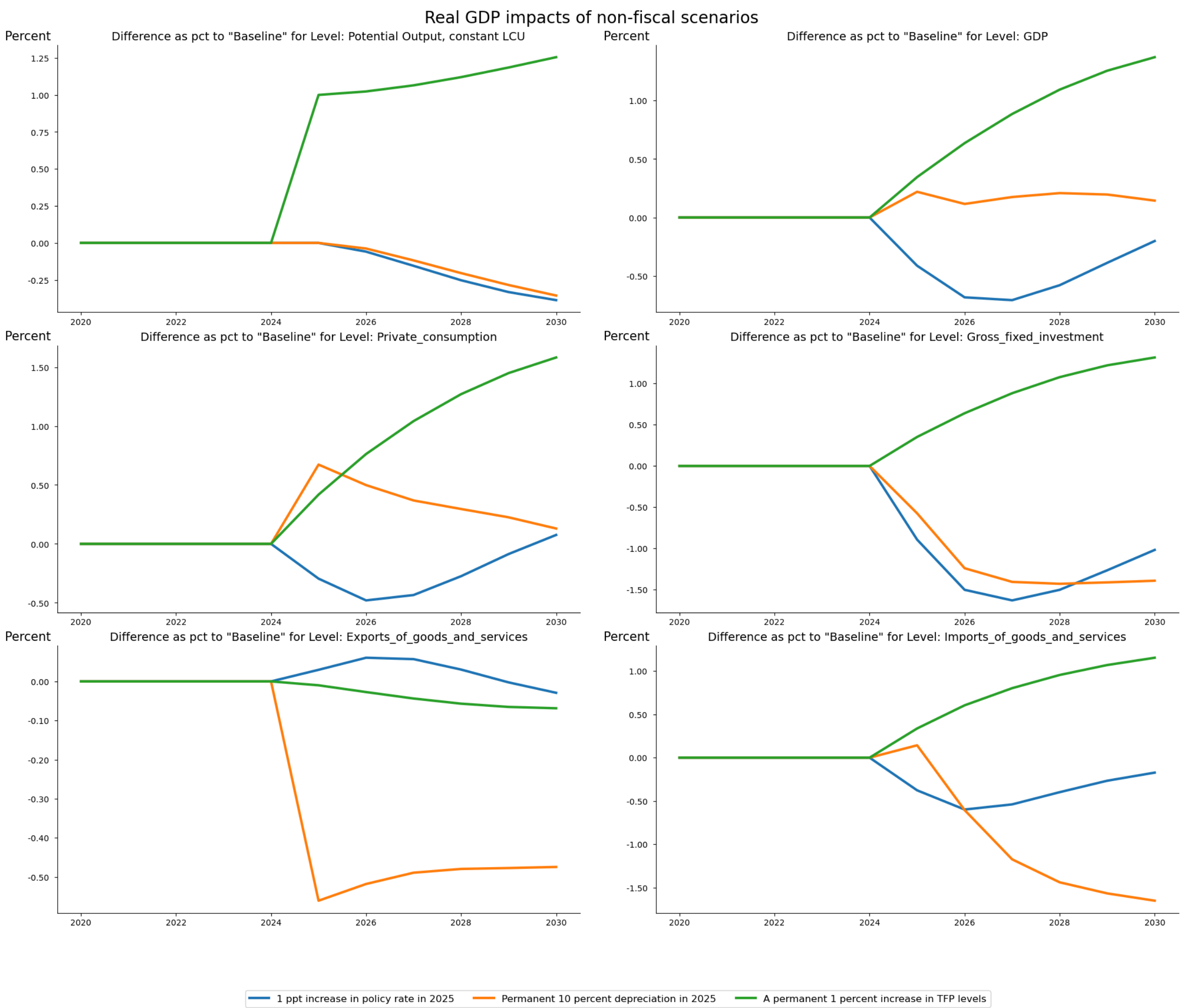
In [49]:

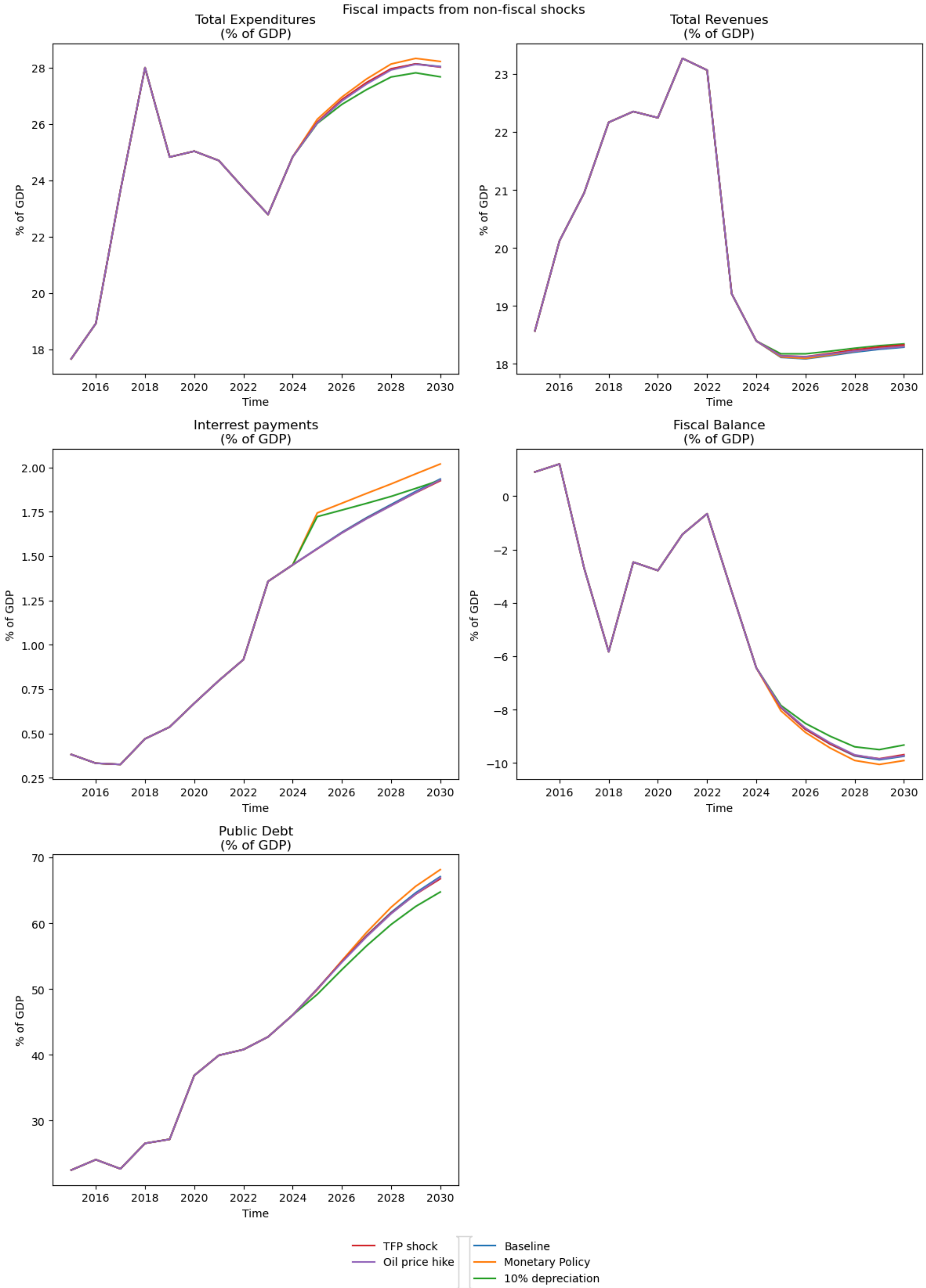
```
for key,df in themodel.keep_solutions.items():
    df[f'{Cty}GGBALOVRLCN_']=df[f'{Cty}GGBALOVRLCN']/df[f'{Cty}NYGDPMKTPCN']*100
    df[f'{Cty}GGEXPTOTLCN_']=df[f'{Cty}GGEXPTOTLCN']/df[f'{Cty}NYGDPMKTPCN']*100
    df[f'{Cty}GGREVTOTLCN_']=df[f'{Cty}GGREVTOTLCN']/df[f'{Cty}NYGDPMKTPCN']*100
    df[f'{Cty}GGEXPINTPCN_']=df[f'{Cty}GGEXPINTPCN']/df[f'{Cty}NYGDPMKTPCN']*100
    df[f'{Cty}GGDBTTOTLCN_']=df[f'{Cty}GGDBTTOTLCN']/df[f'{Cty}NYGDPMKTPCN']*100
    themodel.keep_solutions[key]=df
```

In [50]:

```
base = themodel.keep_solutions['Baseline'].loc[2015:]
MP = themodel.keep_solutions['1 ppt increase in policy rate in 2025'].loc[2015:]
EXR = themodel.keep_solutions['Permanent 10 percent depreciation in 2025'].loc[2015
TFP = themodel.keep_solutions['A permanent 1 percent increase in TFP levels'].loc[2
Oil = themodel.keep_solutions['A permanent $20 increase in global crude oil prices'



figcombo4,axs=plt.subplots(3,2,figsize=(12,16))
i=0

vars=['GGEXPTOTLCN','GGREVTOTLCN','GGEXPINTPCN','GGBALOVRLCN','GGDBTTOTLCN']
for idx,var in enumerate(vars):
    row = idx // 2  # Calculate the row index based on the two-column grid
    col = idx % 2
    axs[row,col].plot(base[f'{Cty}{var}_'],label='Baseline')
    axs[row,col].plot(MP[f'{Cty}{var}_'],label='Monetary Policy')
    axs[row,col].plot(EXR[f'{Cty}{var}_'],label='10% depreciation')
    axs[row,col].plot(TFP[f'{Cty}{var}_'],label='TFP shock')
    axs[row,col].plot(Oil[f'{Cty}{var}_'],label='Oil price hike')
    axs[row,col].set_ylabel('% of GDP')
    axs[row,col].set_xlabel('Time')
```

```python
axs[0,0].set_title('Total Expenditures\n(% of GDP)')
axs[0,1].set_title('Total Revenues\n(% of GDP)')
axs[1,0].set_title('Interrest payments\n(% of GDP)')
axs[1,1].set_title('Fiscal Balance\n(% of GDP)')
axs[2,0].set_title('Public Debt\n(% of GDP)')

#delete the blank chart
figcombo4.delaxes(axs[2, 1])

# Add a joint legend below the subplots with only the first two elements
handles, labels = axs[0,0].get_legend_handles_labels()

#Use only the labels of the first six series in the legend
figcombo4.legend(handles[:6], labels[:6], loc='upper center', bbox_to_anchor=(0.5,
figcombo4.suptitle("Fiscal impacts from non-fiscal shocks")
# Adjust layout to prevent overlapping titles and labels
plt.tight_layout()
plt.show()
```




Fiscal impacts from non-fiscal shocks
Total Expenditures
(% of GDP)
Total Revenues
(% of GDP)
Interrest payments
(% of GDP)
Fiscal Balance
(% of GDP)
Public Debt
(% of GDP)
% of GDP
Time
2016
2018
2020
2022
2024
2026
2028
2030
TFP shock
Oil price hike
Baseline
Monetary Policy
10% depreciation

In [ ]: