

CHEMICKÝ PRŮMYSL

MOŽNOSTI ROZŠÍŘENÍ PROGRAMU PRO SIMULACI CHEMICKÝCH PROCESŮ

JAROSLAV POŽIVIL a JOSEF CHUDOBA

*Ústav počítačové a řídicí techniky, Vysoká škola chemicko-technologická v Praze, Technická 5, 166 28 Praha 6
pozivilj@vscht.cz, chudobaj@vscht.cz*

Došlo 1.4.03, k tisku 15.10.03.

Klíčová slova: univerzální simulační programy, stacionární simulace, reakční kinetika

Obsah

1. Úvod
2. Uživatelské přizpůsobení programu HYSYS
 - 2.1. Technika automatizace (OLE)
 - 2.2. Vytvoření rozšíření ve Visual Basicu nebo ve Visual C++
 - 2.3. Proměnné a jednotkové operace definované uživatelem
3. Příklad rozšíření – katalytická oxidace SO₂ na SO₃
 - 3.1. Reakční kinetika
 - 3.2. Simulační model
 - 3.3. Výsledky a diskuse
4. Rozšiřující reakční kinetika oxidace SO₂
5. Závěr

1. Úvod

HYSYS se v současné době řadí mezi celosvětově nejrozšířenější univerzální simulační programy¹ pro chemické procesy. Přestože jeho součástí je rozsáhlá knihovna jednotkových operací a objemná databáze fyzikálních dat, problematika simulace chemických procesů je natolik složitá, že jsme nuceni posuzovat kvalitu simulačních programů především podle možností rozšířit je o vlastní modely aparátů, o vlastní termodynamické modely, experimentální data, výpočetní metody apod. Typickým problémem je oblast reakční kinetiky, kde jsme většinou odkázáni na výsledky měření a vlastní vztahy. Proto jsme testovali možnosti rozšíření simulačního programu HYSYS.Process v oblasti chemické kinetiky.

2. Uživatelské přizpůsobení programu HYSYS

Přístupy k rozšiřování programu HYSYS je možné rozdělit do tří skupin:

- technika automatizace (OLE),
- vnější programová rozhraní,
- proměnné a jednotkové operace definované uživatelem.

2.1. Technika automatizace (OLE)

Přizpůsobování programu HYSYS potřebám uživatele je usnadněno tím, že HYSYS plně využívá techniku automatizace (Automation). Jde o techniku Microsoft založenou na modelu COM (Component Object Model), který umožňuje jedné aplikaci (programu) pracovat s objekty náležejícími jiným aplikacím. Vyvinula se z techniky spojování a vkládání objektů OLE (Object Linking and Embedding), a proto se někdy používá termín OLE Automation. Technika OLE dovoluje vzít určitý objekt (např. tabulku Excelu) a vložit ho s propojením do jiného objektu (např. textového dokumentu), přičemž změny v tabulce se automaticky promítají do textového dokumentu.

Technika automatizace umožňuje programátorovi zpřístupnit objekty jednoho programu pro užití v jiných aplikacích. V prostředí MS Windows je tato technika již standardem. Jejím principem je sdílení objektů v jednotlivých programech, přičemž tyto objekty jsou definovány na základě reference na zdrojový program, ve kterém jsou vytvořeny. Technika automatizace pracuje způsobem klient – server. Serverem je aplikace, která umožňuje ostatním aplikacím pracovat s jejími soubory. Klientem automatizace je aplikace, která pracuje s objekty jiných aplikací pomocí automatizace. Přitom většina aplikací může tvořit jak klienta automatizace, tak server automatizace.

HYSYS je aplikace serveru automatizace (poskytuje požadované služby), jiný program, např. Visual Basic, představuje klienta automatizace (požaduje služby). Můžeme v něm napsat kód, který má přístup k objektům programu HYSYS a využívá je. Uživatel napíše program podle svých potřeb v libovolném prostředí, které podporuje techniku automatizace (Visual C++, Visual Basic, Delphi, aj.) a využívá pro své účely vlastnosti proudů, fyzikálně-chemické vlastnosti látek či matematické modely simulačního programu HYSYS. Program předřazený simulačnímu programu HYSYS může sloužit i jako maska, která využívá důležité výsledky simulace a skrývá komplikovanost řešení. Typickým příkladem je propojení mezi MS Excelem a programem HYSYS pomocí jazyka VBA (Visual Basic for Application), který je součástí MS Excelu. Propojení Excelu a HYSYS je vhodné například pro prezentaci výsledků analýz citlivosti. Tento doplněk je již k dispozici na internetu² i s příslušným zapracovaným grafickým rozhraním včetně dokumentace.

Architektura klient – server se využívá i opačným směrem – programu HYSYS se zpřístupní rozšiřující moduly, jak popíšeme dále.

2.2. Vytvoření rozšíření ve Visual Basicu nebo ve Visual C++

Funkčnost programu HYSYS verze 2.0 lze rozšířit přidáním následujících uživatelských objektů k simulaci:

- rozšiřující jednotková operace (Unit Operation),
- rozšiřující reakční kinetika (Reaction Kinetics),
- rozšiřující balíček fyzikálně-chemických vlastností (Property Package).

Tato rozšíření se vytvářejí pomocí rozhraní mezi programem HYSYS a vývojovým programovacím prostředím Visual Basic nebo Visual C++ (rozšiřující balíček fyzikálně-chemických vlastností lze vytvořit pouze ve Visual C++). HYSYS obsahuje knihovny objektů, ke kterým mají tyto dva programovací jazyky přístup. Propojení je uskutečněno přes kontejnery HYSYSu (containers), které obsahují objekty a metody, které používá HYSYS a následně pak i vývojové prostředí. Pro uvedené typy rozšíření se využívají tři typy kontejnerů. Vlastní rozšíření se definuje – kóduje se ve vývojovém prostředí a kód je po odladění zkompilován jako knihovna DLL (Dynamic Linked Library). Posledním stupněm je registrace knihovny DLL do systému registrů Windows. I když uvedená tři rozšíření zůstávají v separátním souboru DLL, při používání se tváří jako integrální součást programu HYSYS.

Při vytváření rozšíření se postupuje v šesti stupních. Postup popíšeme pro rozšíření ve Visual Basicu:

1. Vytvoření definičního souboru rozšíření (EDF = extension definition file). Tento soubor obsahuje důležité informace vyžadované kontejnerem rozšíření programu HYSYS, např. jména a typy vlastních proměnných nebo jeho identifikaci (ProgId). Pro tvorbu souboru EDF se používá program dodávaný společně se simulačním programem HYSYS – editor pohledů (Extension View Editor).
2. Vytvoření grafického prostředí objektu (Object View). Tento stupeň sice není povinný, ale umožňuje uživateli interaktivní práci s vytvořeným objektem.
3. Implementace metod požadovaných kontejnerem, např. rozšiřující jednotkové operace musí mít implementovány metody Inicializuj (Initialize) a Proveď (Execute).
4. Registrace rozšíření, přičemž se vyžaduje, aby registrovaný identifikátor ProgId byl stejný jako ten, který jsme vložili do definičního souboru rozšíření EDF.
5. Odladění rozšíření pomocí standardních postupů Visual Basicu. Např. je možno vložit body přerušení, nejlépe do metody Inicializuj, nebo se použije ladicí okno Debug atd.
6. Distribuce rozšíření. Jakmile je rozšíření odladěno, lze vytvořit ActiveX DLL soubor výběrem příslušného příkazu Visual Basicu v nabídce Soubor (File). Výsledkem je rozšíření, které je možné distribuovat, přičemž uživateli zůstanou skryté použité informace či metody. Distribuovat je nutné jak soubor DLL, tak soubor EDF, popř.

další soubory, pokud je rozšíření vyžaduje. Pro spuštění vlastního modulu je třeba mít nainstalovaný soubor DLL a soubor EDF v jednom adresáři.

2.3. Proměnné a jednotkové operace definované uživatelem

Funkčnost simulačního programu HYSYS lze zlepšit použitím proměnných a jednotkových operací definovaných uživatelem. Uživatelem definované proměnné (User Variables) jsou po své definici plnohodnotnými proměnnými, které nelze rozlišit od vestavěných proměnných objektů HYSYS. Lze je používat stejným způsobem v rámci celého objektu simulace (Simulation Case). Uživatelem definované proměnné se píší v interním jazyku maker, který je součástí programu HYSYS. Interní editor maker (Macro Language Editor) je interaktivní prostředí pro vývoj, testování a provádění skriptů. Používá syntax velmi podobnou syntaxi jazyka MS Visual Basic. Umožňuje zápis kódů, které je možné spouštět v programu HYSYS v mnoha různých situacích.

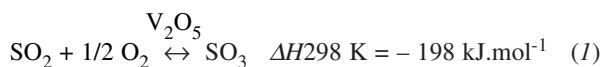
S uživatelem definovanými proměnnými lze např. pracovat v rámci analýz citlivosti, lze je monitorovat graficky a také je exportovat do jiných simulací, neboť každá nová proměnná je uložena do svého souboru. Podobně je tomu také s uživatelem definovanými jednotkovými operacemi. Jde o vlastní modely jednotkových operací, které jsou rovněž programované v interním jazyku maker programu HYSYS.

Příkladem využití může být výpočet Reynoldsova čísla při proudění tekutin, výpočet Wobbeho čísla pro plynné směsi, výpočet rosného bodu nebo vyjádření cenových nákladů proudících médií.

3. Příklad rozšíření – katalytická oxidace SO₂ na SO₃

3.1. Reakční kinetika

K popisu kinetiky reakcí v plynné fázi na tuhém katalyzátoru jsou v programu HYSYS k dispozici modely Langmuira a Hinselwooda, Eleya a Rideala a Marse a Van Krevelena³. Tyto obecné modely ne vždy přesně popisují kinetiku reakcí na určitých katalyzátorech. Z tohoto důvodu je často nutné doplnit simulační úlohu o vlastní model, který vychází z experimentálních měření nebo je publikován v literatuře. Jedním z takových příkladů je oxidace SO₂ na SO₃ na vanadovém katalyzátoru v průmyslově důležité výrobě kyseliny sírové:



Jde se o exotermní vratnou reakci, která probíhá adiabaticky v etážovém reaktoru prakticky za konstantního tlaku. Kinetická data byla získána měřením vzorků průmyslových

katalyzátorů na laboratorním diferenciálním průtokovém reaktoru s recirkulací plynu^{4,5,6}. Pro vyjádření závislosti reakční rychlosti na teplotě byla použita kinetická rovnice^{7,8}

$$r = 0,5 \cdot A \cdot \exp(-E/RT) \cdot (y_{\text{O}_2} \cdot y_{\text{SO}_2}^{0,5} - \frac{y_{\text{SO}_3}}{K_p}) \quad (2)$$

kde r je reakční rychlost, A frekvenční faktor, E aktivační energie, T teplota, y molární zlomek a K_p rovnovážná konstanta reakce (1). Konstanta K_p je pouze funkcí teploty a vychází z experimentálních dat⁹:

$$\log K_p = \frac{4812,3}{T} - 2,8254 \log T + 2,284 \cdot 10^{-3} T - 7,012 \cdot 10^{-7} T^2 + 1,197 \cdot 10^{-10} T^3 + 2,23 \quad (3)$$

3.2. Simulační model

Cílem výpočtů byla simulace chování první etáže adiabatického reaktoru na oxidaci SO_2 . K tomu byl použit knihovní model reaktoru s pístovým tokem (PFR) a knihovní termodynamický model PRSV (modifikace Pengovy a Robinsonovy stavové rovnice). Kinetika reakce – vztahy (2) a (3) byla naprogramována jako rozšíření (Reaction Kinetics Extension) v prostředí Visual Basicu 6.0 a reakce se tak stala součástí nabídky reakcí v programu. Do kinetické rovnice jsme použili experimentální data pro starší typ katalyzátoru:

$$\text{pro } T < 763,34 \text{ K} \quad A = 1,80 \cdot 10^8 \text{ kmol} \cdot \text{kg}^{-1} \cdot \text{h}^{-1}, \\ E = 121,4 \text{ kJ} \cdot \text{mol}^{-1}$$

$$\text{pro } T \geq 763,34 \text{ K} \quad A = 2,55 \cdot 10^4 \text{ kmol} \cdot \text{kg}^{-1} \cdot \text{h}^{-1}, \\ E = 65,3 \text{ kJ} \cdot \text{mol}^{-1}$$

Pro získání názornější představy o náročnosti/jednoduchosti vytváření rozšíření je dále uveden jeho kód ve Visual Basicu. Za apostrofy jsou uvedeny vysvětlující komentáře (všechny znaky uvedené mezi apostrofem a koncem řádku považuje Visual Basic za komentář).

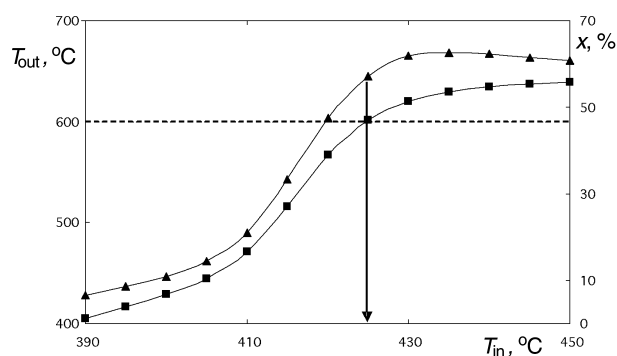
Pro simulaci byla použita reálná průmyslová data první etáže reaktoru, jehož základní údaje jsou uvedeny v tabulce I.

Tabulka I
Parametry simulovaného reaktoru

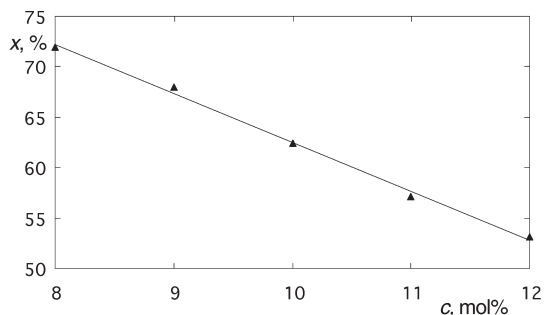
| Parametr | Hodnota |
|--|---------|
| Průměr reaktoru, m | 7,07 |
| Výška vrstvy katalyzátoru v prvním loži, m | 0,495 |
| Sypná hmotnost katalyzátoru, $\text{kg} \cdot \text{m}^{-3}$ | 580 |
| Teplota na vstupu do lože, °C | 425 |
| Tlak na vstupu do lože, kPa | 114,4 |
| Průtok plynu, $\text{kg} \cdot \text{s}^{-1}$ | 10,902 |

Kvůli verifikaci modelu byl nejprve simulován základní výrobní režim – denní výroba 313 t H_2SO_4 při koncentraci SO_2 10,9 mol.% ve vstupním plynu. Protože bylo dosaženo dobré shody s daty získanými z reálného procesu, byl model využit ke zkoumání vlivu některých provozních parametrů na stupeň konverze:

- závislost stupně konverze a výstupní teploty plynu na teplotě vstupujícího plynu,
- závislost stupně konverze na koncentraci SO_2 ve vstupujícím plynu,
- průběh konverze a teploty podél lože katalyzátoru.



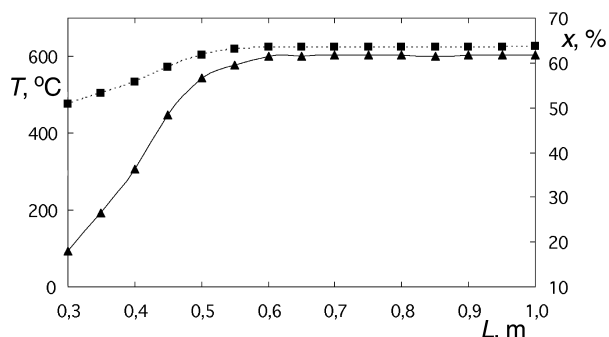
Obr. 1: Závislost stupně konverze SO_2 (x) a výstupní teploty plynu (T_{out}) na teplotě vstupujícího plynu (T_{in});
▲ konverze, ■ teplota



Obr. 2: Závislost stupně konverze SO_2 (x) na koncentraci SO_2 ve vstupujícím plynu (c)

3.3. Výsledky a diskuse

Závislost stupně konverze a výstupní teploty plynu na teplotě vstupujícího plynu je znázorněna v obr. 1. Optimalizace vstupní teploty do vrstvy katalyzátoru představuje u exotermní vratné reakce nejjednodušší možnost racionalizace výroby bez nutných investic. Vstupní teplotu je třeba pružně přizpůsobovat změnám režimu výroby (snižovat při rostoucí koncentraci SO_2 a zvyšovat při rostoucím průtoku plynu) i stavu katalyzátoru (jeho postupná dezaktivace se dá do jisté míry kompenzovat vzrůstem vstupní teploty). Při stanovení optimální vstupní teploty jsme omezeni výstupní



Obr. 3: Závislost stupně konverze SO_2 (x) a teploty (T) na délce lože katalyzátoru; \blacktriangle konverze, \blacksquare teplota

teplotou, která nesmí překročit maximální přípustnou hodnotu, při které dochází k deaktivaci katalyzátoru. Ta se obvykle pohybuje podle použitého typu katalyzátoru mezi 600 a 650 °C, pro určitý starší typ činí 600 °C. Proto optimální vstupní teplota základního režimu nemůže být vyšší než 425 °C a maximální dosažitelná konverze je pak 57,6 %. Závislost stupně konverze na koncentraci SO_2 ve vstupujícím plynu je znázorněna na obr. 2. V grafu jsou vyneseny konverze při optimální vstupní teplotě, která odpovídá maximální přípustné teplotě na výstupu z etáže 600 °C.

Konverze SO_2 je příznivě ovlivňována přebytkem vzduchu, ale přitom současně rostou náklady související s průchodem plynu výrobou, jsou nutná rozměrnější zařízení a zejména klesá využití reakčního tepla, které ohřívá větší množství inertního dusíku. Aby se lépe využila energie pro výrobu páry, dává se proto přednost vyšším koncentracím SO_2 (11–12 %). Při použití dvoustupňové konverze se 4–5 loži katalyzátoru se pro vyšší koncentrace pokles konverze na první etáži reaktoru v dalších ložích do značné míry vyrovná.

Při další simulaci jsme zaznamenali konverzní a teplotní profil lože katalyzátoru v rozmezí 0,3 m až 1,0 m. Jak je zřejmé z obr. 3, je použití lože delšího než 0,5 m neefektivní, protože se konverze přiblížila rovnovážné hodnotě a dále neroste – bylo dosaženo stupně přeměny 0,647 při teplotě 623 °C.

4. Rozšiřující reakční kinetika oxidace SO_2

Rozšíření simulačního programu HYSYS je ilustrováno příkladem reakční kinetiky katalytické oxidace oxidu siřičitého při výrobě kyseliny sírové. Potřebná kinetická rovnice byla naprogramována jako rozšíření v prostředí Visual Basicu 6.0 a celá reakce se tak stala součástí nabídky reakcí v programu.

```
Option Explicit      'vsechny promenne musi byt deklarovany;
                    'usnadnuje to jejich identifikaci a zabranuje chybam pri preklepu
Dim hyContainer As Object      'deklarace globalnich objektu HYSYSU
                    'predpona hy oznacuje promenne pochazejici z HYSYSu

Public Function Initialize(ByVal Container As Object, ByVal IsRecalling As Boolean)
As Long
    'Tato funkce se vola, kdyz je rozsireni poprve pridano do programu nebo kdyz je
    nahrano
On Error GoTo chyba      'v pripade chyby ukonci program
Dim hyReactant As Reactant      'Deklarace lokalnich objektu v HYSYSU
Initialize = extnCurrentVersion      'overeni pouzivane verze HYSYSU
Set hyContainer = Container      'Odkaz na kontejner, který obsahuje metody pro za-
cleneni rozsireni
    'IsRecalling je False pri prvni pridani rozsireni k simulaci
If IsRecalling = False Then
    'reakcni podminky:
    hyContainer.Phase = ptVapourPhase      'plynna reakcni faze
    hyContainer.ReactionBasis = rbMoleFracBasis      'molarni zlomky
    hyContainer.Reactants.RemoveAll      'odtrani vseh prvodnich reaktantu
    'pridani pozadovanych reaktantu a jejich stechiometrie:
    Set hyReactant = hyContainer.Reactants.Add(„SO2“)
    hyReactant.StoichiometricCoefficientValue = -1
    hyContainer.BaseReactant = hyReactant      'klicova slozka v rychlostni reakci
    Set hyReactant = hyContainer.Reactants.Add(„Oxygen“)
    hyReactant.StoichiometricCoefficientValue = -0.5
    Set hyReactant = hyContainer.Reactants.Add(„SO3“)
    hyReactant.StoichiometricCoefficientValue = 1
    'nedefinovane vlastnosti se nastavi jako vypoctene takze je nelze me-
    nit:
```

```

With hyContainer
    .SetReactionPropertyState rpReactants, vsCalculated
    .SetReactionPropertyState rpStoichiometricCoefficients, vsCalculated
    .SetReactionPropertyState rpMinTemperature, vsCalculated
    .SetReactionPropertyState rpMaxTemperature, vsCalculated
    .SetReactionPropertyState rpReactionBasis, vsCalculated
    .SetReactionPropertyState rpReactionPhase, vsCalculated
    .SetReactionPropertyState rpBaseReactant, vsCalculated
    .SetReactionPropertyState rpBasisConversion, vsCalculated
    .SetReactionPropertyState rpRateConversion, vsCalculated
End With
End If
chyba: 'navesti pro chybu
End Function

Public Function ReactionRate(ByVal Fluid As Object, ByVal RxnTemperatureInC As
Double, ByVal RxnVolumeInKmolPerM3 As Double, rate As Double) As Boolean
    'rychlost reakce (funkce ReactionRate) se vola pri kazdem vypoctu pomoci
    rozsireni
    On Error GoTo chyba 'zachyceni moznych chyb

    'definice lokalnich promennych
    Dim TK As Double 'reakcni teplota v Kelvinech
    Dim KSO3 As Variant 'rovnovazna konstanta
    Const Ro As Long = 580 'sypna hustota katalyzatoru (kg/m3)
    Const R As Single = 8.314 'univerzalni plynova konstanta J/K*mol
    'indexy slozek - pro prehlednost programu:
    Dim SO2Index As Integer
    Dim O2Index As Integer
    Dim SO3Index As Integer
    Dim A As Double 'predexponencialni Arrheniuv faktor
    Dim B As Double 'aktivacni energie
    Dim ComponentFrac As Variant
    'molarni zlomky slozek:
    Dim ySO2 As Double
    Dim ySO3 As Double
    Dim yO2 As Double

    TK = RxnTemperatureInC + 273.15 'reakcni teplota v Kelvinech
    'experimentalni hodnoty aktivacni energie a predexponencialniho faktoru
    If TK >= 763.34 Then
        A = 25500# / 3600 'kmol/kg/s
        B = 15600 * 4.186 ' J/mol
    Else
        A = 180000000# / 3600 'kmol/kg/s
        B = 29000 * 4.186 'J/mol
    End If
    'vypocet rovnovazne konstanty jako funkce teploty
    KSO3 = 10 ^ (4812.2 / TK - 2.8254 / 2.3025 * Log(TK) + 0.002284 * TK -
0.0000007012 * TK ^ 2 + 0.0000000001197 * TK ^ 3 + 2.23)
    'indexovani slozek (na zaklade vlastnosti HYSYS Containeru)
    SO2Index = Fluid.Components.Index(„SO2“)
    SO3Index = Fluid.Components.Index(„SO3“)
    O2Index = Fluid.Components.Index(„Oxygen“)
    ComponentFrac = Fluid.MolarFractionsValue 'molarni slozeni vstupniho proudu
    ySO2 = ComponentFrac(SO2Index)
    ySO3 = ComponentFrac(SO3Index)
    yO2 = ComponentFrac(O2Index)

```

```

'reakcni rychlost (kmol/kg kat./s)
rate = 0.55 * A * Exp(-B / (R * TK)) * (yO2 * ySO2 ^ 0.5 - ySO3 / KSO3)
' prepocet na vnitřní jednotky HYSYSu:
rate = rate * Ro      'kmol/s/m3
ReactionRate = True  'reakcni rychlost spočtena
Exit Function
chyba:      'reakcni rychlost není spočtena
ReactionRate = False
End Function

```

5. Závěr

Simulace chemických procesů je natolik složitá, že ani rozsáhlé univerzální simulační programy nejsou dostatečně univerzální, a tak je kvalita prostředků pro jejich rozšíření významným kritériem při jejich hodnocení. Typickou oblastí je reakční kinetika, kde jsme většinou odkázáni na výsledky měření a vlastní vztahy. Vzhledem k otevřené architektuře poskytuje program HYSYS.Process několik možností rozšíření, které jsou podle našeho názoru dostatečné. Jejich využívání však není zcela triviální a vyžaduje vzhledem k objektovému charakteru programu speciální znalosti a alespoň minimální zkušenost s používáním objektově orientovaného programovacího jazyka, jako je např. Visual Basic.

Tato práce vznikla za finanční podpory projektu MŠMT ČR (výzkumného záměru) MSM 223400007.

LITERATURA

1. *Simulation and Model Based Control Software Industry Outlook, 1996–2001*, compiled by Automation Research Corporation, Dedham, MA; Software Usage Surveys by 100 % Software Solutions, Inc, and the University of Edinburgh.
2. www.hyprotech.com, staženo 14.5.2002
3. HYSYS Process 2.2 Documentation. 2000 AEA Technology – Hyprotech Ltd. [CD ROM]

4. Vosolsobě J., Michálek J.: Chem. Listy 60, 87 (1966).
5. Michálek J., Vosolsobě J.: Chem. Prum. 17, 527 (1967).
6. Michálek J., Vosolsobě J.: Chem. Prum. 30, 461 (1980).
7. Michálek J., Vosolsobě J., Muchlenov I. P.: Ž. prikl. chim. 48, 80 a 766 (1975).
8. Michálek J., Vosolsobě J.: Sborník VŠCHT Praha B19, 55 a 67 (1975).
9. Twigg M. V.: *Catalyst Handbook*, 2. vyd., I. Manson Publishing, London 1996.

J. Poživil and J. Chudoba (*Department of Computer and Control Instrumentation, Institute of Chemical Technology, Prague*): **Possibilities of Program Extension for Simulation of Chemical Processes**

Simulation of chemical processes is insofar complex that even the latest simulation programs are not sufficiently versatile and so the quality of means for their extension is a significant criterion for their evaluation. The article deals with the extension of the HYSYS simulation program. A typical field is reaction kinetics where we mostly depend on measurement results and own relationships. The implementation of experimental models of reaction kinetics into program is shown on the example of catalytic oxidation of SO₂ to SO₃. The described possibilities of program extension make it a versatile tool for the use in simulation of chemical processes.